

AD-A194 339

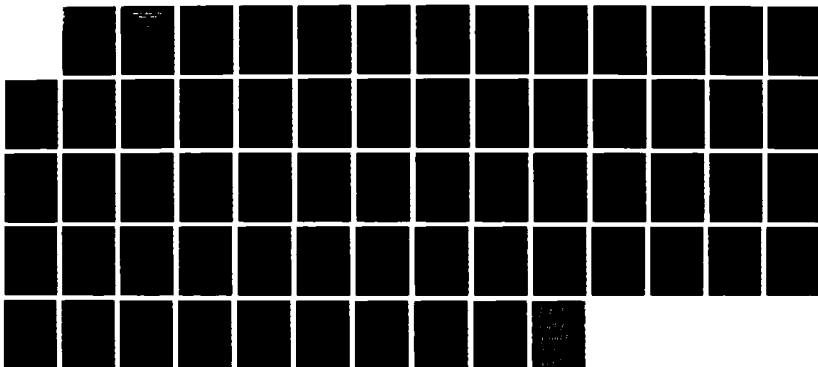
ESTIMATION FORCES ACTING ON AN UNDERWATER VEHICLE WITH
GPS (GLOBAL POSITIONING SYSTEM) AND KALMAN FILTERING
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J K EASTON

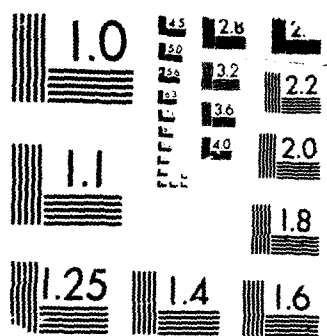
1/1

UNCLASSIFIED

MAR 88

F/O 13/10.1 NL





MICROCOPY RESOLUTION TEST CHART
 (NBS 1963-A)

DTIC FILE COPY

②

NAVAL POSTGRADUATE SCHOOL Monterey, California

AD-A194 339



THESIS

ESTIMATING FORCES ACTING ON AN
UNDERWATER VEHICLE WITH
G.P.S. AND KALMAN FILTERING

by

James K. Easton, Jr.

March 1988

Thesis Advisor:

H.A. Titus

Approved for public release; distribution is unlimited.

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution Availability of Report		
2b Declassification Downgrading Schedule			Approved for public release; distribution is unlimited.		
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)		
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol (if applicable) 32	7a Name of Monitoring Organization Naval Postgraduate School		
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000			7b Address (city, state, and ZIP code) Monterey, CA 93943-5000		
8a Name of Funding Sponsoring Organization		8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number		
8c Address (city, state, and ZIP code)			10 Source of Funding Numbers		
			Program Element No	Project No	Task No
			Work Unit Accession No		
11 Title (include security classification) ESTIMATING FORCES ACTING ON AN UNDERWATER VEHICLE WITH G.P.S. AND KALMAN FILTERING					
12 Personal Author(s) James K. Easton Jr.					
13a Type of Report Master's Thesis		13b Time Covered From To		14 Date of Report (year, month, day) March 1988	15 Page Count 64
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Cosau Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	underwater vehicle, navigation, Kalman Filter		
19 Abstract (continue on reverse if necessary and identify by block number)					
A discrete state space model is developed which describes an autonomous underwater vehicle and incorporates the effects of currents, sea state, and wind as it travels through the sea. Heading commands are calculated to overcome these effects by the design of an Extended Kalman Filter which estimates their combined velocity components. Simulations are done which test the filter's effectiveness in a range of different environments. Some potential uses for this system are discussed at the end.					
20 Distribution Availability of Abstract			21 Abstract Security Classification		
<input checked="" type="checkbox"/> unclassified unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users			Unclassified		
22a Name of Responsible Individual Harold A. Titus			22b Telephone (include Area code) (408) 646-2560		22c Office Symbol 32Ts

DD FORM 1473,84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Estimating Forces Acting on an Underwater
Vehicle with G.P.S. and Kalman Filtering

by

James K. Easton Jr.
Lieutenant, United States Navy
B.S.E.E., University of Idaho, 1979


Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
ELECTRICAL ENGINEER


from the

NAVAL POSTGRADUATE SCHOOL
March 1988

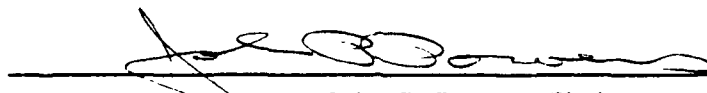
Author:

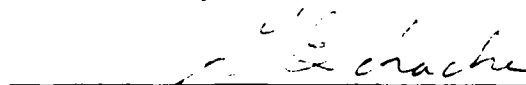

James K. Easton Jr.

Approved by:


Harold A. Titus, Thesis Advisor

Roberto Cristi, Second Reader


John P. Powers, Chairman,
Department of Electrical and Computer Engineering


Gordon E. Schacher,
Dean of Science and Engineering

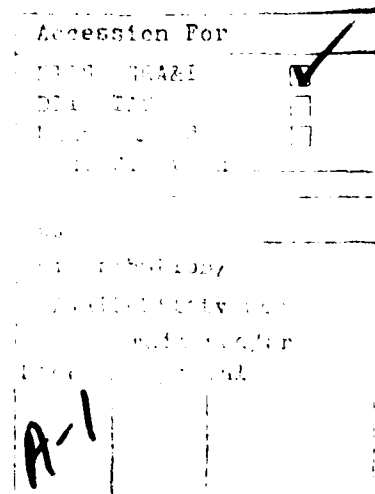


TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. GLOBAL POSITIONING SYSTEM	3
C. GPS RECEIVER	5
II. SYSTEM MODEL	6
A. DYNAMIC MODEL	6
B. STATE SPACE REPRESENTATION	10
III. STATE ESTIMATION	13
A. INTRODUCTION	13
B. KALMAN FILTER	13
1. The Linear Kalman Filter	13
2. Extended Kalman Filter	14
C. MATRIX CALCULATIONS	15
1. Determination of Q'	15
2. Determination of R	18
3. Determination of $\hat{x}_{k-1 k}$	18
4. Determination of H_k	18
IV. SIMULATION RESULTS	21
A. INTRODUCTION	21
B. STATE ESTIMATION WITH NO BIAS	23
C. STATE ESTIMATION WITH BIAS AND COURSE CORRECTION ...	28
D. STATE ESTIMATION WITH MULTIPLE SET AND DRIFT CHANGES	33
E. STATE ESTIMATION WITH MEASUREMENT NOISE	38
V. CONCLUSION	44
A. ANALYSIS	44
B. APPLICATIONS	44
1. Back up to an IMU	44

2. Mine Location	45
3. Long Range Reconnaissance	45
C. RECOMMENDATIONS FOR FUTURE RESEARCH	45
APPENDIX A. SIMULATION PROGRAM	46
APPENDIX B. KALMAN FILTER PROGRAM	51
LIST OF REFERENCES	54
INITIAL DISTRIBUTION LIST	55

LIST OF FIGURES

Figure 1.	System Block Diagram	2
Figure 2.	Navigation Message Contents	4
Figure 3.	Velocity Components Encountered by the Vehicle	7
Figure 4.	Dynamic Model [Adapted from Ref. 2]	8
Figure 5.	Step Response for Vehicle	9
Figure 6.	Symbol Identification	10
Figure 7.	Summary of Extended Kalman Equations	16
Figure 8.	Kalman Filter Recursive Loop	17
Figure 9.	Intended and Actual Vehicle Tracks With No Added Bias	24
Figure 10.	Position Coordinates With No Added Bias	25
Figure 11.	Velocity Components With No Added Bias	26
Figure 12.	Set and Drift Components With No Added Bias	27
Figure 13.	Intended and Actual Vehicle Tracks With Bias and Course Correction	29
Figure 14.	Position Coordinates With Bias and Course Correction	30
Figure 15.	Velocity Components With Bias and Course Correction	31
Figure 16.	Set and Drift Components With Bias and Course Correction	32
Figure 17.	Intended and Actual Vehicle Tracks With Multiple Set and Drift Changes	34
Figure 18.	Position Coordinates With Multiple Set and Drift Changes	35
Figure 19.	Components With Multiple Set and Drift Changes	36
Figure 20.	Set and Drift Components With Multiple Set and Drift Changes	37
Figure 21.	Intended and Actual Vehicle Tracks With Measurement Noise	40
Figure 22.	Position Coordinates With Measurement Noise	41
Figure 23.	Velocity Components With Measurement Noise	42
Figure 24.	Set and Drift Components With Measurement Noise	43

LIST OF TABLES

Table 1.	SUMMARY OF VALUES USED IN SIMULATION RUNS	22
Table 2.	DIFFERENCES IN POSITION COORDINATES	38
Table 3.	DIFFERENCES IN VELOCITY COMPONENTS	38
Table 4.	DIFFERENCES IN SET AND DRIFT COMPONENTS	38

I. INTRODUCTION

A. BACKGROUND

With the increased costs for Manned Naval Surface and Subsurface assets and the greatly increased capability of small computing devices, using small autonomous underwater vehicles (AUVs) to carryout certain missions has become increasingly desirable. A vehicle's capability to navigate to a precise location independently and in many cases covertly is essential.

Current research into potential navigation systems that measure a vehicle's position and unknown forces acting on it include sonar doppler, strapdown Inertial Measuring Units (IMUs), and sonar triangularization. The two primary IMUs underdevelopment for AUVs are the Laser Ring Gyro and the Fiber Optic Gyro (FOG). The systems mentioned above each have their own independent drift characteristics which have to be filtered out and their actual positions updated from time to time using an outside source such as the Navstar Global Positioning System (GPS) or Omega. In addition, they have large power requirements, are bulky (with the possible exception of the FOG), and require complex control circuitry. If a microprocessor could be programmed to estimate the underwater forces acting on the vehicle enroute to the destination, based on updates by GPS, and then to apply a heading command to compensate for any deviations in its course, then the power requirements, complexity, and cost could be significantly reduced. Figure 1 on page 2 illustrates a block diagram of a possible system.

In this Thesis, a state space model will be developed based on a generic vehicle's dynamics. The forces acting on the vehicle will be modeled as a velocity vector with an angle (Drift) and a magnitude (Set) with the following assumptions:

- the angle and magnitude of this vector are white random Gaussian variables with zero mean about predicted values, and
- the overall Set and Drift between sample intervals remains a constant.

In order to estimate this set and drift, an extended Kalman Filter is developed. Various simulations are performed which explore the behavior of the filter and its ability to estimate the velocity components. The simulation was done using DSL (Dynamic Simulation Language) on the Naval Postgraduate School's IBM 3033 Mainframe Computer.

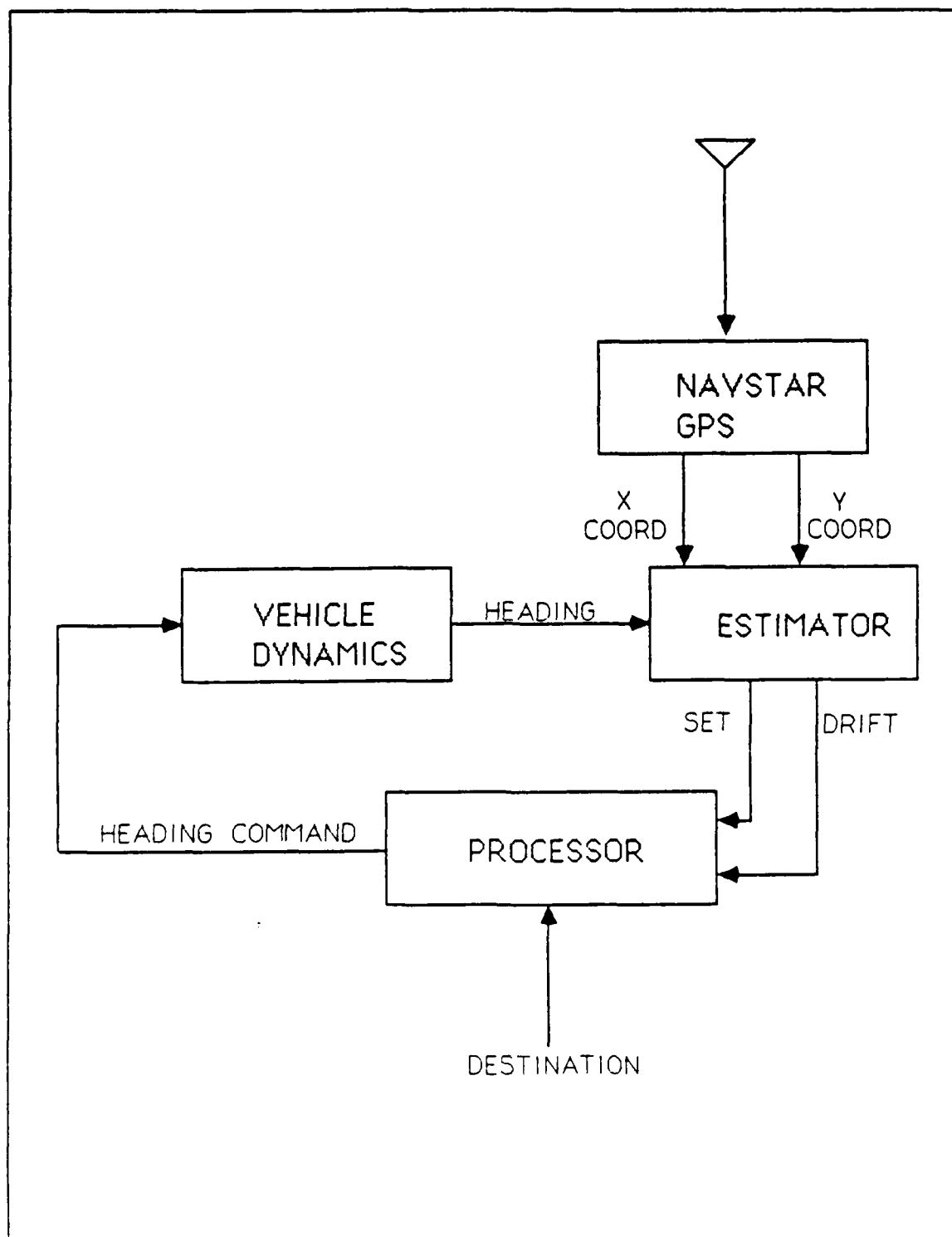


Figure 1. System Block Diagram

B. GLOBAL POSITIONING SYSTEM

The Navstar Global Positioning System is a space-based radio navigation system which operates passively. GPS is a tri-service multiagency program managed by the Air Force's GPS Joint Program Office. When fully operational in the early 1990's, the system will consist of 18 satellites with three orbiting spares, in 6 orbital planes. Each satellite transmits two unique codes, the clear acquisition (C/A) code and the Precise (P) code. These codes, which are unique to each satellite, are modulated and two L Band frequencies, L1 and L2, as pseudo random noise sources along with the Navigation Message. L1 contains both the C/A and P codes while L2 has just the P code.

The user's receiver generates the identical codes and correlates its code with the satellite's. The amount of slewing that the receiver must do in order to obtain a match is the time that the two signals are off. Dividing by the speed of light and correcting for atmospheric delays and any clock differences produces the range to the satellite. Obtaining three ranges from three different satellites is required to obtain a 3 dimensional coordinate position fix. However, in order to accomplish this, the receiver's clock would have to be identical in time to that of the GPS's Master Clock which would defeat the purpose of the system. To provide this time differential, the receiver must lock onto 4 satellites and solve four simultaneous equations for the three coordinates and the time differential in order to get the fix.

The GPS Navigation Message supplies the user with the satellite's clock offset, atmospheric delays, and satellite ephemeris in order to accurately solve the four equations. It is modulated on both L1 and L2 at a 50 bit per second rate and is contained in a 1500 bit frame. This frame is divided into 5 subframes which contain specific blocks of data. This format is shown in Figure 2 on page 4.

The first subframe contains the clock correction parameters and ionospheric propagation delay model parameters. The second and third subframes contain the satellite's ephemeris which is updated every 1-5 hours by the ground control segment of GPS. Subframe four is for any special messages which must be passed onto the users.

The final subframe contains the almanac data. The almanac data is an abbreviated form of the first three subframes for each of the satellites. This data is transmitted on a rotating basis and requires 19 subframes to transmit it all and is used only to help locate the four satellites which are overhead. The almanac must be loaded into the receiver either manually or by the receiver itself by locking onto a satellite and obtaining the block five message through an entire cycle.

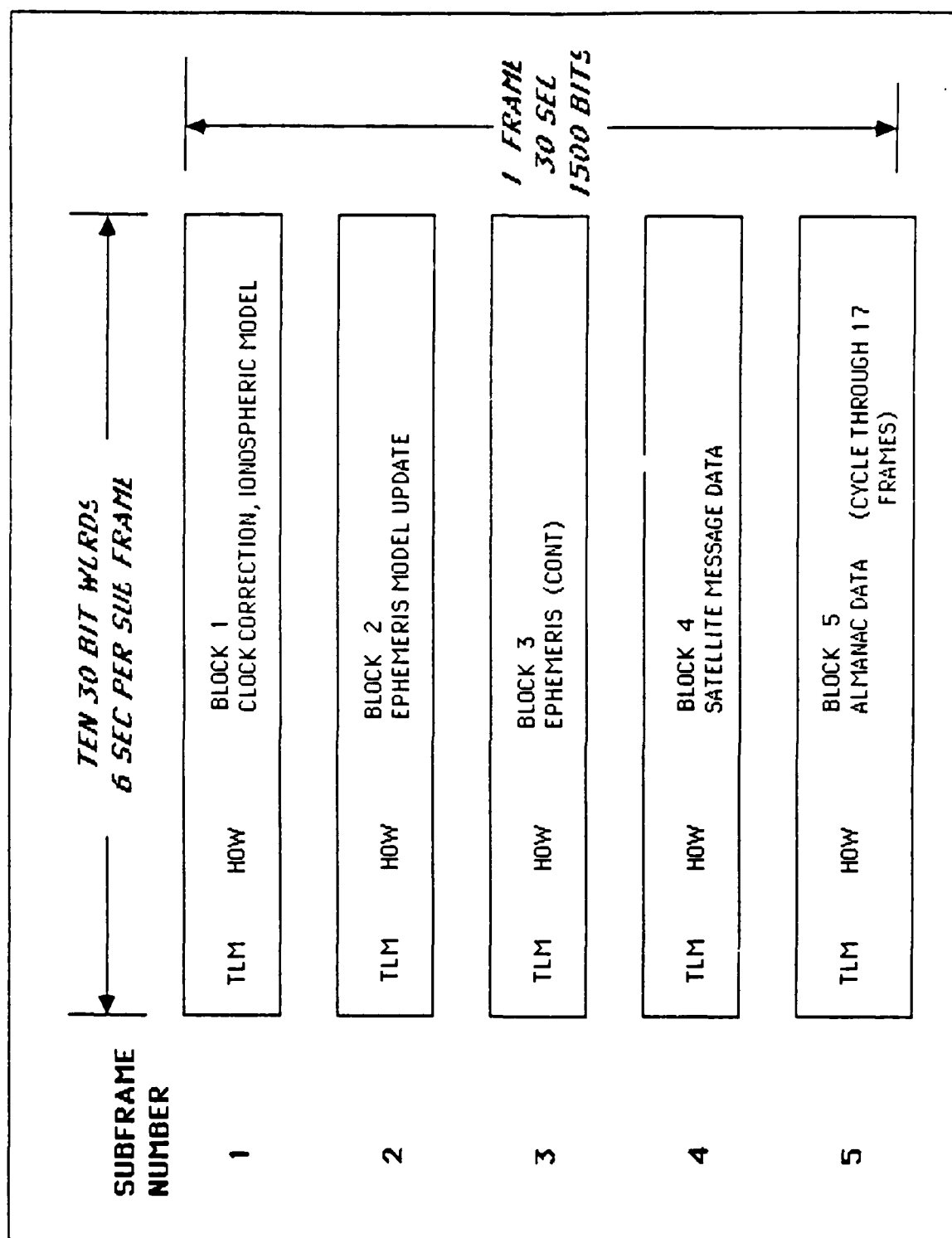


Figure 2. Navigation Message Contents

C. GPS RECEIVER

In order to perform the tasks that will be referred to later in this thesis, the GPS receiver must be capable of accomplishing the following tasks:

- Obtain current position
- Store the necessary waypoints, including the start and end points.
- Provide course information to next waypoint.
- Have the ability to survive temperature, pressure, and other conditions associated with the marine environment.

Currently under development for the Marine Corps by the Naval Ocean Systems Command (NOSC) San Diego is a receiver called the Small Unit Navigation System (SUNS). [Ref.1] This receiver is designed to be handheld, rugged, lightweight, and battery operated for the Marine Corp's Reconnaissance teams during long range operations on both land and underwater. SUNS will be able to store up to 100 waypoints, including the destination, and will provide the user with current position, velocity, cross track error, distance and bearing to next waypoint, and time with a positional accuracy of 50 ft.

One additional requirement which should be considered is equipping the receiver with a multi-channel capability to lock on all four satellites at the same time. This would allow the vehicle to obtain a position fix quicker and thus spend less time on or near the surface. The SUNS receiver is being designed as a single channel unit and will have to sequentially correlate each of the signals from the four satellites.

II. SYSTEM MODEL

A. DYNAMIC MODEL

An underwater vehicle will travel through the water with a desired heading and at an ordered speed. The combined external forces acting on the vehicle will cause it to deviate from the intended track at some angle (Drift) and velocity (Set). This deviation can be modeled as a single velocity vector when added to the vehicle's own velocity vector results in the True Velocity vector along which the vehicle will travel. This relationship is shown in Figure 3 on page 7, where α is the vehicle's commanded heading, γ is the angle for the True Velocity, and ψ is the angle of the drift's velocity. The simple addition of the individual components produce Equations 2.1 and 2.2 for the x and y components of V_t , x_T and y_T .

$$\dot{x}_T = Vel \cos(\alpha) + Set \cos(\psi) \quad (2.1)$$

$$\dot{y}_T = Vel \sin(\alpha) + Set \sin(\psi) \quad (2.2)$$

Developed in Reference 2, the vehicle's dynamic model, neglecting any sway dynamics, is illustrated in Figure 4 on page 8. In this model, a heading input (Hdg) produces a heading direction (α), whose sine and cosine are multiplied by the vehicle's ordered speed (Vel) to produce the ordered velocity components (x and y). If these components are summed with their respective drift components (ux and uy), then the components of the true velocity (x_t and y_t) are obtained.

$G(s)$ is the transfer function describing the vehicle's true dynamics, which for the generic vehicle used in this thesis will be a simplified first order system. With $k = 4.4$ and $G(s) = \frac{1}{s + 4}$, a transient response as shown in Figure 5 on page 9 is realized. HDGCOM is the applied heading command and ALPHA is the resultant vehicle heading angle (α). This $G(s)$ was chosen as a reduced order model based on various vehicles which are now in operation. $K = 4.4$ produces a 2 sec. settling time with little overshoot. This type of response will minimize the effect of the vehicle's response to heading commands on the Kalman Estimator updating during the simulations in Chapter 4 and is close to what a real vehicle would produce.

As can be seen from the model, this is an open loop system where the heading angle is put in from an outside source and maintained by the position feedback of α . This

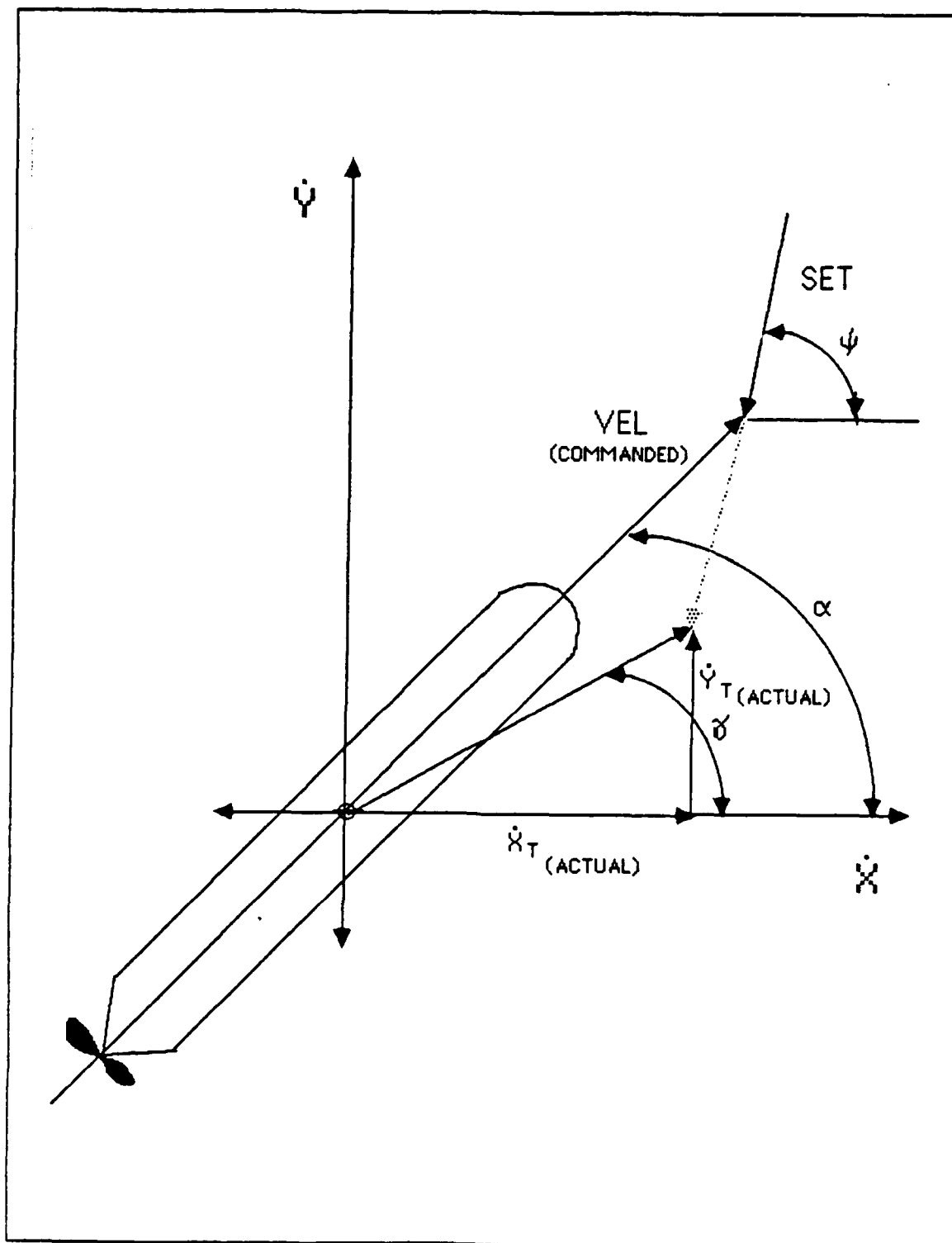


Figure 3. Velocity Components Encountered by the Vehicle

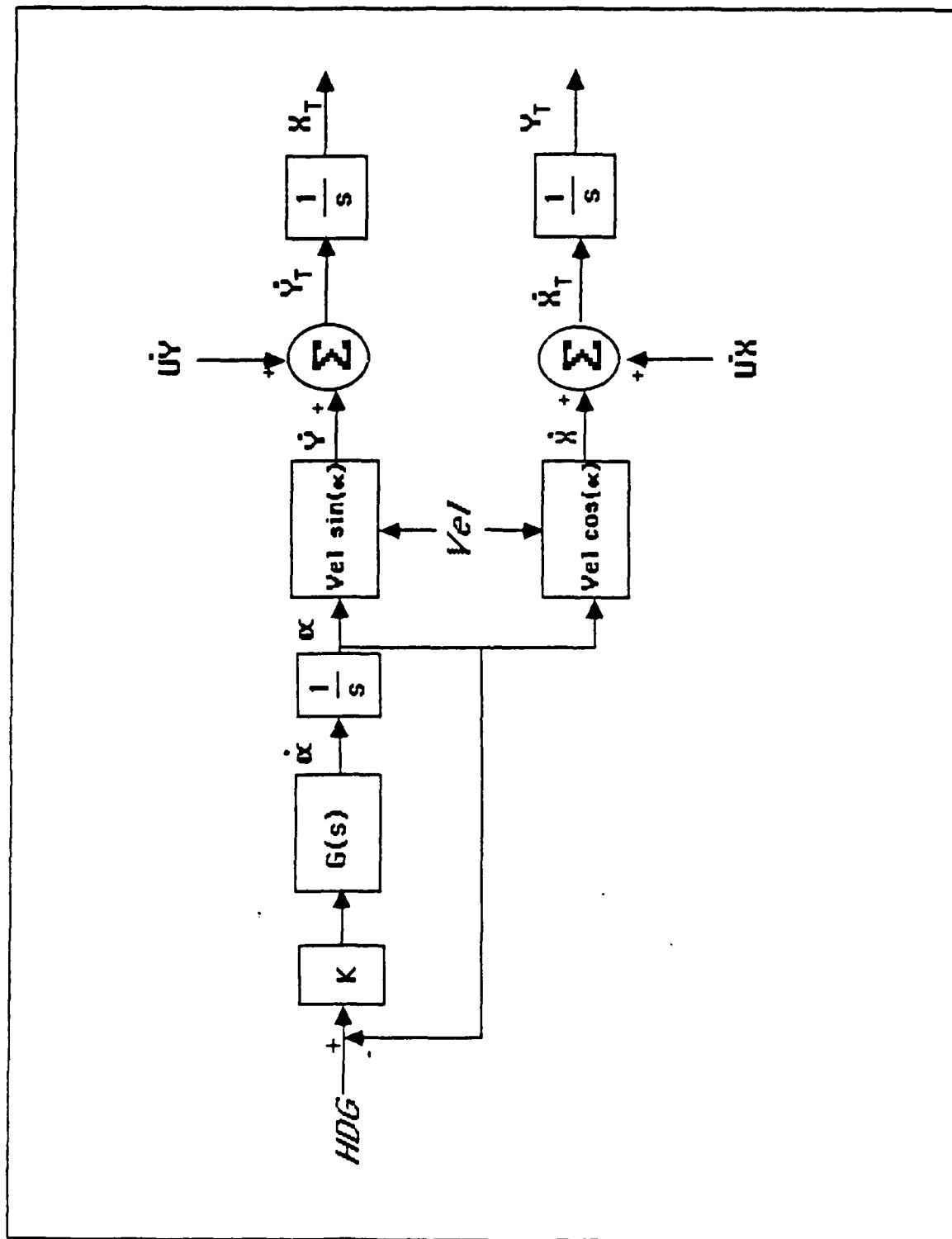


Figure 4. Dynamic Model [Adapted from Ref. 2 : p 19]

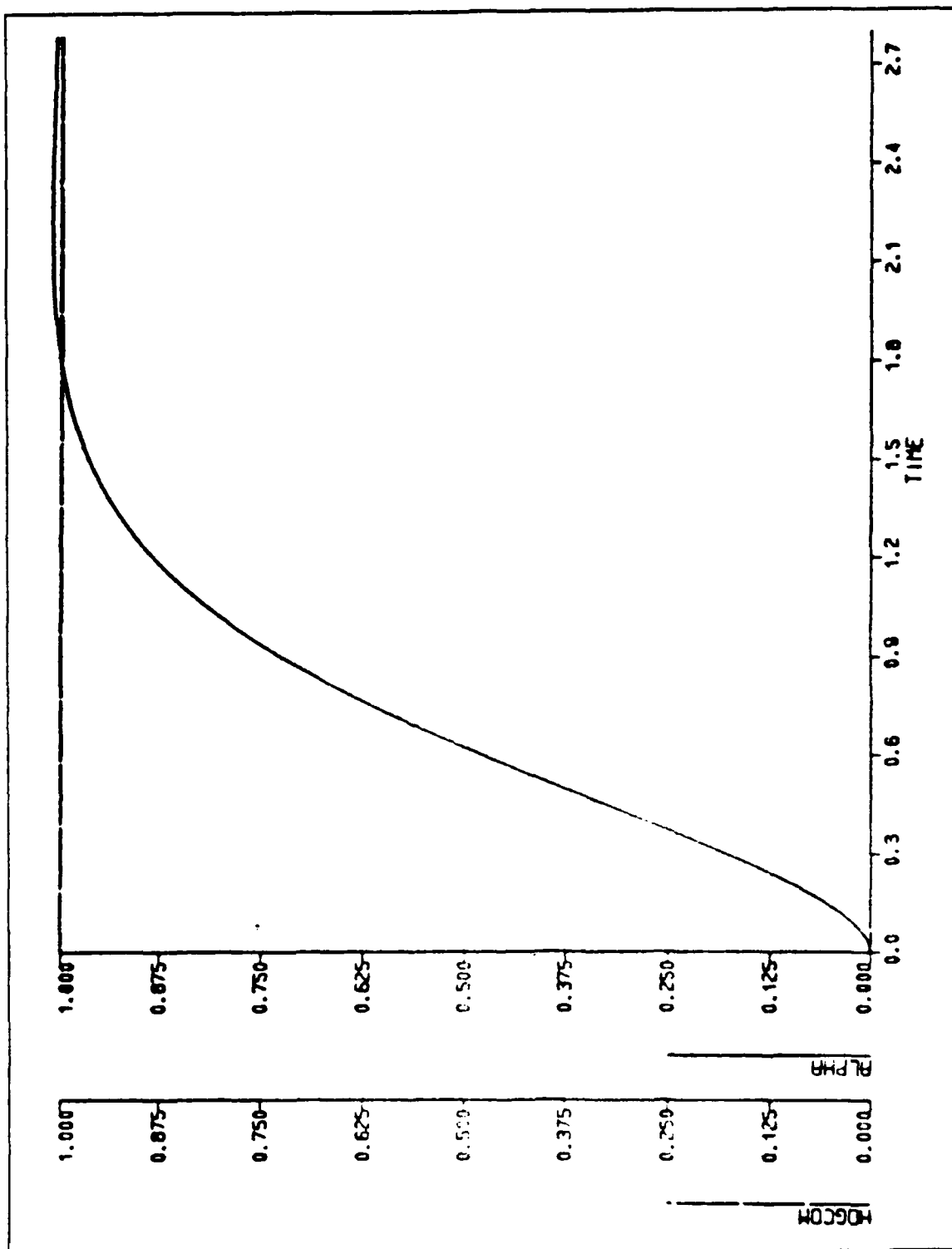


Figure 5. Step Response for Vehicle

heading command will be provided by the processor after calculating u_x and u_y and determining an α which will negate these inputs.

B. STATE SPACE REPRESENTATION

In this section a discrete state space model is developed based on the dynamic model above along with the identification of the individual states.

The linear discrete state space model is described by Equations 2.3 and 2.4 [Ref.3: pp. 66-67] where \underline{x}_k is the current value of the states at time $= t$ and \underline{x}_{k-1} contains the predicted values of the states at time $= t + T$. T is defined as the sampling period. The definitions for the variables in the equations below are summarized in Figure 6.

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma_1 \underline{u}_k + \Gamma_2 \underline{w}_k \quad (2.3)$$

$$\underline{z} = H \underline{x}_k + \underline{v}_k \quad (2.4)$$

$\underline{x} \equiv N \times 1$ state vector
 $\underline{u}_k \equiv M \times 1$ vector of applied forcing functions
 $\underline{w}_k \equiv K \times 1$ vector of random forcing functions
 $\underline{z} \equiv L \times 1$ observation vector
 $\underline{v} \equiv L \times 1$ measurement noise vector
 $\Phi \equiv N \times N$ Transition matrix
 $\Gamma_1 \equiv N \times M$ matrix
 $\Gamma_2 \equiv N \times K$ matrix
 $H \equiv L \times N$ matrix

Figure 6. Symbol Identification

The following states were defined and placed into \underline{x} :

$$x(1) = x_t \quad (2.5a)$$

$$x(2) = \dot{x}_t \quad (2.5b)$$

$$x(3) = y_t \quad (2.5c)$$

$$x(4) = \dot{y}_t \quad (2.5d)$$

$$x(5) = \dot{u}x \quad (2.5e)$$

$$x(6) = \dot{u}y \quad (2.5f)$$

The Φ Matrix becomes

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

The command vector, \underline{u}_k , will consist of the velocity components, ux_k and uy_k , of any new heading given to the vehicle which are needed to compensate for the estimated set and drift. They are given in the equations below with Hdg_k and Hdg_{k-1} as the heading commands at time = kT and $(k-1)T$ respectively.

$$ux_k = Vel \times (\cos(Hdg_k) - \cos(Hdg_{k-1})) \quad (2.7)$$

$$uy_k = Vel \times (\sin(Hdg_k) - \sin(Hdg_{k-1})) \quad (2.8)$$

Γ_1 now becomes:

$$\Gamma_1 = \begin{bmatrix} T & 0 \\ 1 & 0 \\ 0 & T \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.9)$$

The x and y components of the Set and Drift are contained in \underline{w}_k as w_x and w_y . Since the Set's magnitude and the Drift's direction are assumed to be white Gaussian random variables with zero mean about predicted values, w_x and w_y will also be white Gaussian random variables with zero mean about their predicted values. This leads to Γ_2 below.

$$\Gamma_2 = \begin{bmatrix} T & 0 \\ 1 & 0 \\ 0 & T \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.10)$$

At this point, the observation vector, z , consists only of the position coordinates (x and y) which come from GPS. The velocity components received from GPS are those components that the vehicle is experiencing at that instant in time and may or may not be the time averaged velocities which are defined in the model. If only the position coordinates are used then the system is not observable [Ref. 3: pp 67-70]. As a result another variable must be measured.

In the system model (Figure 4 on page 8), recall that α is fed back to the input in order to maintain the applied heading command. Since this value must be measured, it can serve as the other variable that will link the states. Thus the observation vector becomes:

$$z = \begin{bmatrix} x_t \\ y_t \\ \alpha \end{bmatrix} + v \quad (2.11)$$

It can be observed that z is now nonlinear because α is not a linear function of the states $x(2)$, $x(4)$, $x(5)$, and $x(6)$. [Eq. 2.12] As a result, it cannot be treated as the linear function as described in Equation 2.4.

$$\alpha = \arctan \left[\frac{x(4) - x(6)}{x(2) - x(5)} \right] \quad (2.12)$$

The next step is to create an estimator that will measure ux and uy in the presence of noise (uncertainty) in the measurements and overcome nonlinear problem of the observation vector. In Chapter 3, this is accomplished by the design of an extended Kalman Filter.

III. STATE ESTIMATION

A. INTRODUCTION

In this chapter, the theory of statistical parameter estimation is applied to determining the values of the Set and Drift components, u_x and u_y , which were discussed in Chapter 2. Because of the nonlinear measurement vector and the uncertainty generated in the measurement noise, a simple observer, i.e., the Luenberger Observer, is not feasible. Due to the successful applications of the Extended Kalman Filter [Ref. 4: p. 23] in solving nonlinear estimation problems and the advancements in microprocessor design in obtaining real time solutions, the Extended Kalman Filter was chosen as the estimator for this problem.

B. KALMAN FILTER

1. The Linear Kalman Filter

The Kalman Filter is an algorithm for extracting or estimating information from noisy data. More specifically, it is a parametric estimation process based on an autoregressive (AR) model of the plant. The filter uses samples or measurements of the plant as observables to recursively update estimates of the plant's state.

When both the system and measurement models are linear functions of the states. The state update equation is shown below in Equations 3.1 and 3.2. In these equations G is the vector containing the optimum estimation gains (Kalman gains), $\hat{x}_{k|k-1}$ is the predicted value of the states at time $= kT$ based on the measurements obtained and any known forcing functions at time $= (k-1)T$, and \hat{x}_k is the updated state estimation vector based on the new measurement vector (z_k).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + G(z_k - H\hat{x}_{k|k-1}) \quad (3.1)$$

The following assumptions are made in using the Kalman equations:

- The random forcing function (w_k) is zero mean and uncorrelated with covariance Q'_k .

$$\begin{aligned} E[w_k] &= 0, \text{ for all } k \geq 0 \\ E[w_k w_j] &= \begin{cases} Q'_k \delta_{kk} & \text{for all } k = j \\ 0 & \text{for all } k \neq j \end{cases} \end{aligned} \quad (3.2)$$

- The measurement noise (v_k) is zero mean and uncorrelated with covariance R_k .

$$E[\underline{v}_k] = 0, \text{ for all } k \geq 0 \quad (3.3)$$

$$E[\underline{v}_k \underline{v}_j^T] = \begin{cases} R_{kk} \delta_{kk} & \text{for all } k = j \\ 0 & \text{for all } k \neq j \end{cases}$$

- The random forcing function and measurement noise are uncorrelated.

$$E[\underline{w}_k \underline{v}_j^T] = E[\underline{v}_k \underline{w}_j^T] = 0 \text{ for all } k, j \neq 0 \quad (3.4)$$

- The random forcing function and initial states are uncorrelated.

$$E[\underline{w}_k \underline{x}_0^T] = E[\underline{x}_0 \underline{w}_k^T] = 0 \text{ for all } k \neq 0 \quad (3.5)$$

- The measurement noise and initial states are uncorrelated.

$$E[\underline{v}_k \underline{x}_0^T] = E[\underline{x}_0 \underline{v}_k^T] = 0 \text{ for all } k \neq 0 \quad (3.6)$$

The optimal estimation gains are those which satisfy the following relationships:

$$K_k = P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R]^{-1} \quad (3.7)$$

$$P_{k|k} = [I - G_k H_k] P_{k|k-1} \quad (3.8)$$

$$P_{k+1|1} = \Phi P_{k|k} \Phi^T + \Gamma_2 Q' \Gamma_2^T \quad (3.9)$$

where $P_{k|k}$ is the error covariance matrix update at time $= k T$ and $P_{k+1|1}$ is the predicted error covariance at time $= (k+1)T$ based on the data at time $= k T$.

2. Extended Kalman Filter

As previously mentioned, the measurement vector as described in Equations 2.11 and 2.12 is a nonlinear function of the states and as a result will be modeled as shown below,

$$z_k = h(\underline{x}_k, k) + \underline{v}_k \quad (3.10)$$

In this model, the measurement vector, z_k , is a function of the state variables plus the measurement noise vector, \underline{v}_k . The adaptation of the Kalman Filter involves the linearizing of $h(\underline{x}_k, k)$ by taking the Taylor Series expansion about the predicted value of the states, $\underline{x}_{k|k-1}$, at time $= k T$ and keeping only the first order terms. This is the Extended Kalman Filter. The reader is referred to Gelb [Ref. 3: pp 180-225] for a detailed explanation of the process. In summary, a more precise approximation can be achieved in the optimal estimator by including higher terms of the series expansion but at the expense of simplicity and increasing the possibility of the recursion diverging.

The linearized form of Equation 3.10 yields

$$\bar{z}_k = H_k \bar{x}_k + v_k \quad (3.11)$$

$$H_k = \left. \frac{\partial h(\bar{x}_k, k)}{\partial \bar{x}_k} \right|_{\bar{x}_k = \hat{\bar{x}}_{k-1}} \quad (3.12)$$

A summary of the definitions for the Extended Kalman Filter estimator for the linear state model [Eq. 2.3] and the linearized measurement model [Eq. 3.11] is displayed in Figure 7 on page 16. A block diagram that illustrates the algorithm for recursively computing the filter is shown in Figure 8 on page 17.

C. MATRIX CALCULATIONS

1. Determination of Q'

Q' as defined in Equation 3.2 is the covariance matrix for the random forcing function of the set and drift velocity components. It is a measure of the uncertainty contained in the predictions. As stated previously, the set and drift experienced by the vehicle is assumed to be Gaussian with 0 mean about the predicted values and as a result their x/y components are independent and will be approximately Gaussian with 0 mean with the standard deviations equal to the maximum expected deviation in the x and y directions. Thus Q'

$$Q' = \begin{bmatrix} \sigma_{wx}^2 & 0 \\ 0 & \sigma_{wy}^2 \end{bmatrix} \quad (3.16)$$

The method used as an approximation to calculate σ_{wx}^2 and σ_{wy}^2 , is the following:

- Determine the maximum amount of variation which can be expected in the set and drift.
- Calculate the cartesian components, x and y, from the values determined above. Set these values equal to $3\sigma_{x,y}$
- Divide the x and y components by 3, subtract them from their respective predicted components, and square to obtain σ_{wx}^2 and σ_{wy}^2 .

As an example, let the maximum expected deviation for the drift be $\pm 20^\circ$ and the maximum expected deviation for the set be $\pm .5$ knots. The predicted values for the Set and Drift are 45° and 1.0 knots. Using the above procedure, $\sigma_{wx}^2 = 0.43$ and $\sigma_{wy}^2 = 0.43$.

$$\begin{aligned} \text{SYSTEM MODEL: } \underline{x}_{k+1} &= \Phi \underline{x}_k + \Gamma_1 \underline{u}_k + \Gamma_2 \underline{w}_k \\ &\text{where } \underline{w}_k \propto N[0, Q'] \end{aligned} \quad (2.3)$$

$$\begin{aligned} \text{MEASURE MODEL: } \underline{z}_k &= h(\underline{x}_{k,k}) + \underline{v}_k \\ &\text{where } \underline{v}_k \propto N[0, Q'] \end{aligned} \quad (3.10)$$

$$\text{INITIAL CONDITIONS: } \underline{x}_0 \propto N[\hat{\underline{x}}_0, P_0] \quad (3.13)$$

$$\text{OTHER ASSUMPTIONS: } E[\underline{w}_k \underline{v}_j^T] = 0 \text{ for all } k, j \geq 0. \quad (3.4)$$

$$\begin{aligned} \text{GAIN EQUATION:} \\ G_k &= P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R]^{-1} \end{aligned} \quad (3.7)$$

$$\begin{aligned} \text{ERROR COVARIANCE UPDATE:} \\ P_{k|k} &= [I - G_k H_k] P_{k|k-1} \end{aligned} \quad (3.8)$$

$$\begin{aligned} \text{STATE ESTIMATION UPDATE EQUATION} \\ \hat{\underline{x}}_{k|k} &= \hat{\underline{x}}_{k|k-1} + G_k [\underline{z}_k - h(\hat{\underline{x}}_{k|k-1})] \end{aligned} \quad (3.1)$$

$$\begin{aligned} \text{ERROR COVARIANCE PROPAGATION EQUATION:} \\ P_{k+1|k} &= \Phi P_{k|k} \Phi^T + Q \end{aligned} \quad (3.9)$$

$$\begin{aligned} \text{STATE ESTIMATION PROPAGATION:} \\ \hat{\underline{x}}_{k+1|k} &= \Phi \hat{\underline{x}}_{k|k} + \Gamma_1 \underline{u}_k \end{aligned} \quad (3.16)$$

$$\text{DEFINITIONS: } H_k = \left. \frac{\partial h(\underline{x}_k, k)}{\partial \underline{x}_k} \right|_{\underline{x}_k = \hat{\underline{x}}_{k|k-1}} \quad (3.12)$$

$$Q = \Gamma_2 Q' \Gamma_2^T \quad (3.14)$$

$$h(\hat{\underline{x}}_{k|k-1}) = h(\underline{x}_{k,k})|_{\underline{x}_{k,k} = \hat{\underline{x}}_{k|k-1}} \quad (3.15)$$

Figure 7. Summary of Extended Kalman Equations

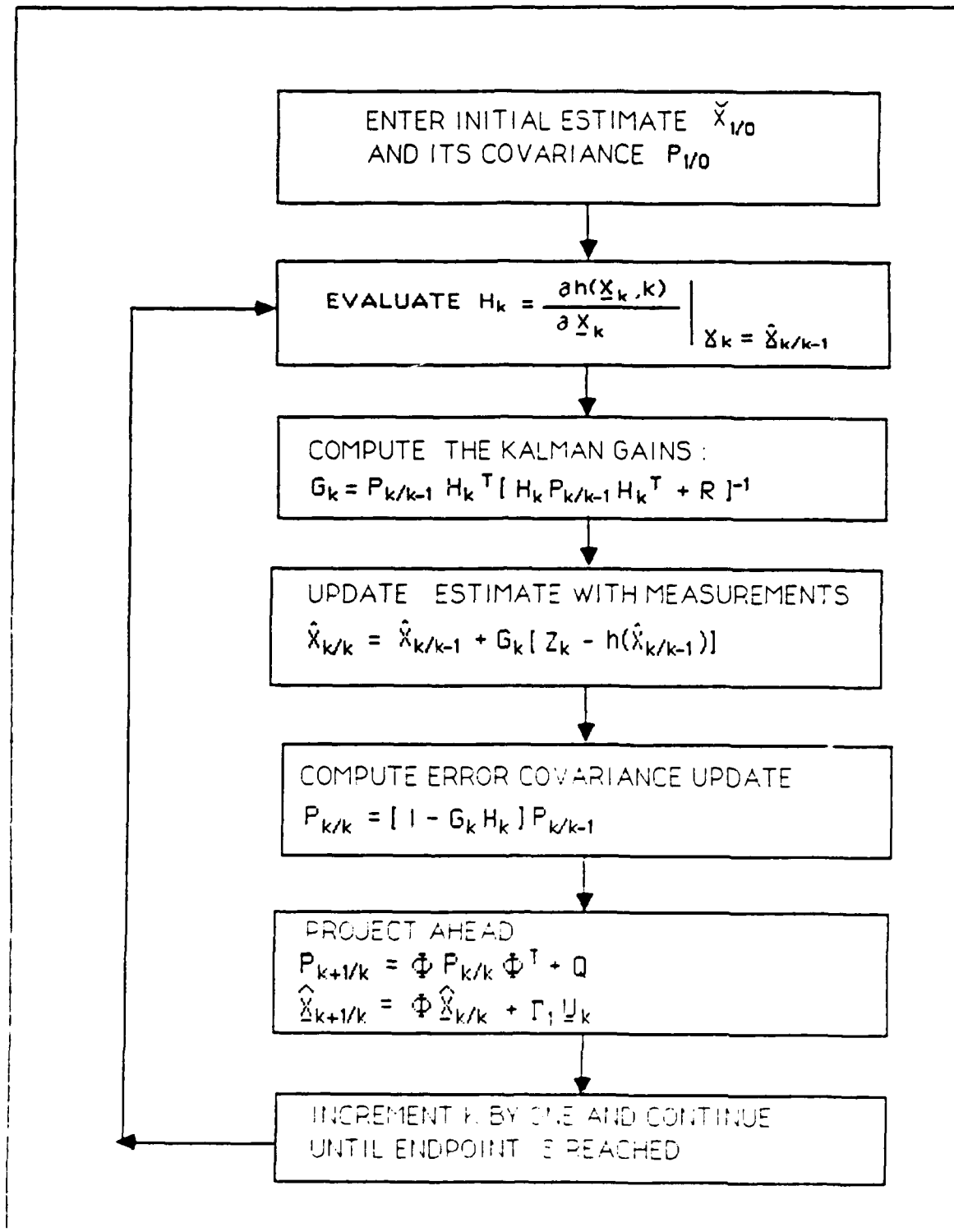


Figure 8. Kalman Filter Recursive Loop

2. Determination of R

R as defined in Equation 3.3, is the covariance matrix for the measurement noise vector. As in Q , R is the uncertainty contained in the measuring of the three parameters x , y , and α in the presence of noise. For this application, the noise for all three parameters is assumed to be uncorrelated amongst themselves and white Gaussian with 0 mean. R becomes

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \quad (3.17)$$

The variances, σ_x , and σ_y , are the standard deviations from GPS and are obtained by using the stated DOD value for the Circular Error of Probability (CEP) of 10 meters or whichever CEP figure is associated with the GPS Receiver in use and Equation 3.18 [Ref. 5: p. 21]. The standard deviation for the specific type of compass used is represented by σ_α .

$$CEP = 0.59(\sigma_x + \sigma_y) \quad (3.18)$$

3. Determination of $\hat{x}_{k+1|k}$

$\hat{x}_{k+1|k}$ as defined in Equation 3.1 is a predictor of the states at time $= k + 1$ based on the current estimation of the states at time $= k$. As a result, $\hat{x}_{k+1|k}$ has to take into account any controls which are applied as a new heading command to overcome the estimated set and drift as well as the known plant dynamics. Thus $\hat{x}_{k+1|k}$ becomes

$$\hat{x}_{k+1|k} = \Phi \hat{x}_k + \Gamma_1 u_k \quad (3.19)$$

where Γ_1 , Φ , and u_k were defined in Equation 2.3.

4. Determination of H_k

H_k as defined in Equation 3.12 is the linearized observation vector and is obtained by taking the partial derivative of z with respect to x . Recalling Equation 2.11, the first two values of z , x and y , are linear functions of only one state and their partial derivatives are easily computed in Equations 3.20 and 3.21.

$$\mathbf{z} = \begin{bmatrix} x_t \\ y_t \\ \arctan \frac{x(4) - x(6)}{x(2) - x(5)} \end{bmatrix} \quad (2.11)$$

$$\frac{\partial z(1)}{\partial \mathbf{X}} = \frac{\partial x}{\partial \mathbf{X}} = [1 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (3.20)$$

$$\frac{\partial z(2)}{\partial \mathbf{X}} = \frac{\partial y}{\partial \mathbf{X}} = [0 \ 0 \ 1 \ 0 \ 0 \ 0] \quad (3.21)$$

To obtain the partial derivative of $z(3)$, the following relationships are used.
[Ref. 6: pp 328-330]

$$\frac{d \arctan(u)}{d x} = \frac{1}{1 + u^2} \frac{d u}{d x} \quad (3.22)$$

$$\frac{d u^{-1}}{d x} = \frac{-1}{u^2} \frac{d u}{d x} \quad (3.23)$$

Using Equations 3.21 and 3.22 and simplifying, the following results were achieved for $\frac{\partial Z(3)}{\partial \mathbf{X}}$:

$$\frac{\partial z(3)}{\partial x(1)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(1)} = 0 \quad (3.24)$$

$$\frac{\partial z(3)}{\partial x(2)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(2)} = \frac{x(2) - x(5)}{(x(2) - x(5))^2 + (x(4) - x(6))^2} \quad (3.25)$$

$$\frac{\partial z(3)}{\partial x(3)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(3)} = 0 \quad (3.26)$$

$$\frac{\partial z(3)}{\partial x(4)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(4)} = \frac{x(4) - x(6)}{(x(2) - x(5))^2 + (x(4) - x(6))^2} \quad (3.27)$$

$$\frac{\partial z(3)}{\partial x(5)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(5)} = \frac{x(5) - x(2)}{(x(2) - x(5))^2 + (x(4) - x(6))^2} \quad (3.28)$$

$$\frac{\partial z(3)}{\partial x(6)} = \frac{\partial \arctan \frac{x(4) - x(6)}{x(2) - x(5)}}{\partial x(6)} = \frac{x(6) - x(4)}{(x(2) - x(5))^2 + (x(4) - x(6))^2} \quad (3.29)$$

The above equations are combined into Equation 3.30, which is the matrix version for \underline{H}_k . The final version, it should be noted, contains only three major calculations: the denominator term which is common, the $x(2) - x(5)$ term, and the $x(4) - x(5)$ term. The two terms, H_{35} and H_{36} are just the negatives of H_{32} and H_{34} respectively. Thus this linearization is not as ominous as one would expect and can easily be implemented on a microprocessor.

$$\underline{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & H_{32} & 0 & H_{34} - H_{32} - H_{34} & 0 & 0 \end{bmatrix} \quad (3.30)$$

where

$$H_{32} = \frac{x(2) - x(5)}{(x(2) - x(5))^2 + (x(4) - x(5))^2} \quad (3.30a)$$

$$H_{34} = \frac{x(4) - x(6)}{(x(2) - x(5))^2 + (x(4) - x(5))^2} \quad (3.30b)$$

IV. SIMULATION RESULTS

A. INTRODUCTION

In this Chapter, the Extended Kalman Filter calculated in Chapter 3 is tested by performing various simulations using the models developed in Chapter 2 and determining if the filter is correctly estimating the six states. This testing was done in four steps. In the first step, the filter and program were checked to insure that the filter accurately estimated the states and that the program interacted with the filter while maintaining the vehicle's track with no added bias. Next, a bias was added to the predicted values of Set and Drift and a heading change is made to correct for the bias and allow the vehicle to continue to its destination. This test would show whether the filter would accept the heading change and maintain its lock. Third, the filter's ability to handle multiple Set and Drift changes is checked by randomly changing the bias every other sample. Last, random noise is added to the measurements being fed into the filter and the errors measured.

The full simulation program which was used to run the last simulation is shown in Appendix A. The other simulations were run with this program with the unnecessary segments commented out. The program primarily consists of Input, Initialization, Dynamic, State Estimation, and Correction Segments. In the Input, the values for the vehicle's speed, the origin and destination coordinates, the predicted values for the Set and Drift, and the standard deviations for the various random variables are entered. The units used for the distance and angle inputs are nautical miles and degrees.

The Initialization segment the nautical miles, hours, and degree units are converted to feet, seconds, and radian units of measure. In addition, equidistant waypoints and course and distance to destination are calculated, biases are added to the predicted Set and Drift, and the Kalman Filter matrices are initialized.

The Dynamic segment calculates the actual x and y coordinates and the current heading angle which are used as the GPS and compass inputs into the filter by using the model of Figure 4 on page 8 created in Chapter 2.

The State Estimation updates \hat{L}_k , $h(\hat{X}_{k-1})$, and \hat{z} and then calls the subroutine Kalman [Appendix B]. Kalman performs the matrix operations and returns the current state estimate, $\hat{X}_{k,k}$, the projected covariance of error, $P_{k-1,k}$, and the projected state estimate, $\hat{X}_{k-1,k}$. Finally the returned estimates of ux and uy are used to calculate the new

heading command in the Course Correction segment which, in addition, updates $\hat{x}_{k-1,k}$ [Eq. 3.16].

For all the simulations, the vehicle's ordered speed or speed through the water is maintained as a constant. This assumes that the vehicle has an optimum speed which minimizes the amount of fuel consumed during a long distance voyage, which appears to be the trend in current research. The program could easily be modified for the vehicle to maintain a constant over the ground speed. The speed picked for all the runs was 7.5 knots.

The predicted Set and Drift were held as a constant for all the runs at 45 degrees and 1.0 knots. The σ_{wx} and σ_{wy} values were those calculated in the example on Page 15 as .66. For σ_x and σ_y , the values used were determined using Equation 3.18 and the DOD figure of 10 meters for the CEP. Thus, $\sigma_x = \sigma_y = 25.4$ feet. The value for σ_v , chosen at random, was 1.5 degrees or .03 radians. A summary of the constants and their values are summarized in Table 1.

Table 1. SUMMARY OF VALUES USED IN SIMULATION RUNS

CONSTANT	VALUE / UNITS
T	600. Seconds
Set	7.5 Knots
Drift	45 Degrees
Orign Coordinates	0.0,0.0 Nautical Miles
Destination Coordinates	7.0,6.0 Nautical Miles
σ_x	25.4 feet
σ_y	25.4 feet
σ_v	.03 radians
σ_{wx}	.66 feet per second
σ_{wy}	.66 feet per second

For each simulation run, there are 7 graphs as output. The first one shows the vehicle's overall track as compared to the desired track using the waypoints as a reference. The remainder are plots of the current state estimates shown against the actual states as a function of time. The values for the actual states are graphed as solid lines, while the estimated state graphs use a dashed line. For convenience to the reader, the actual

number of figures are reduced by combining two plots into each figure with the exception of the vehicle's overall position track.

B. STATE ESTIMATION WITH NO BIAS

In this simulation, the predicted Set and Drift are entered into the vehicle and it is instructed to maintain its course to the destination point. There is no added bias to the Set and Drift and the run is performed for 6000 seconds or 5 sampling periods.

Figure 9 on page 24 shows the vehicle's actual track against the intended track. The waypoints are shown as a reference. As can be seen, the vehicle follows the intended track with no deviation which demonstrates that the program itself is functioning properly. Figure 10 on page 25 provides the estimated and actual position state variables plotted both as a function of time and each other. In Figure 11 on page 26, the true velocity state variables, \dot{x} and \dot{y} , are again plotted as functions of time and estimated vs. actual. Figure 12 on page 27 plots the estimated and actual Set and Drift components, u_x and u_y in the same manner as before. These figures demonstrate that the filter is locking on after only one sample and maintains the lock throughout the run as expected.

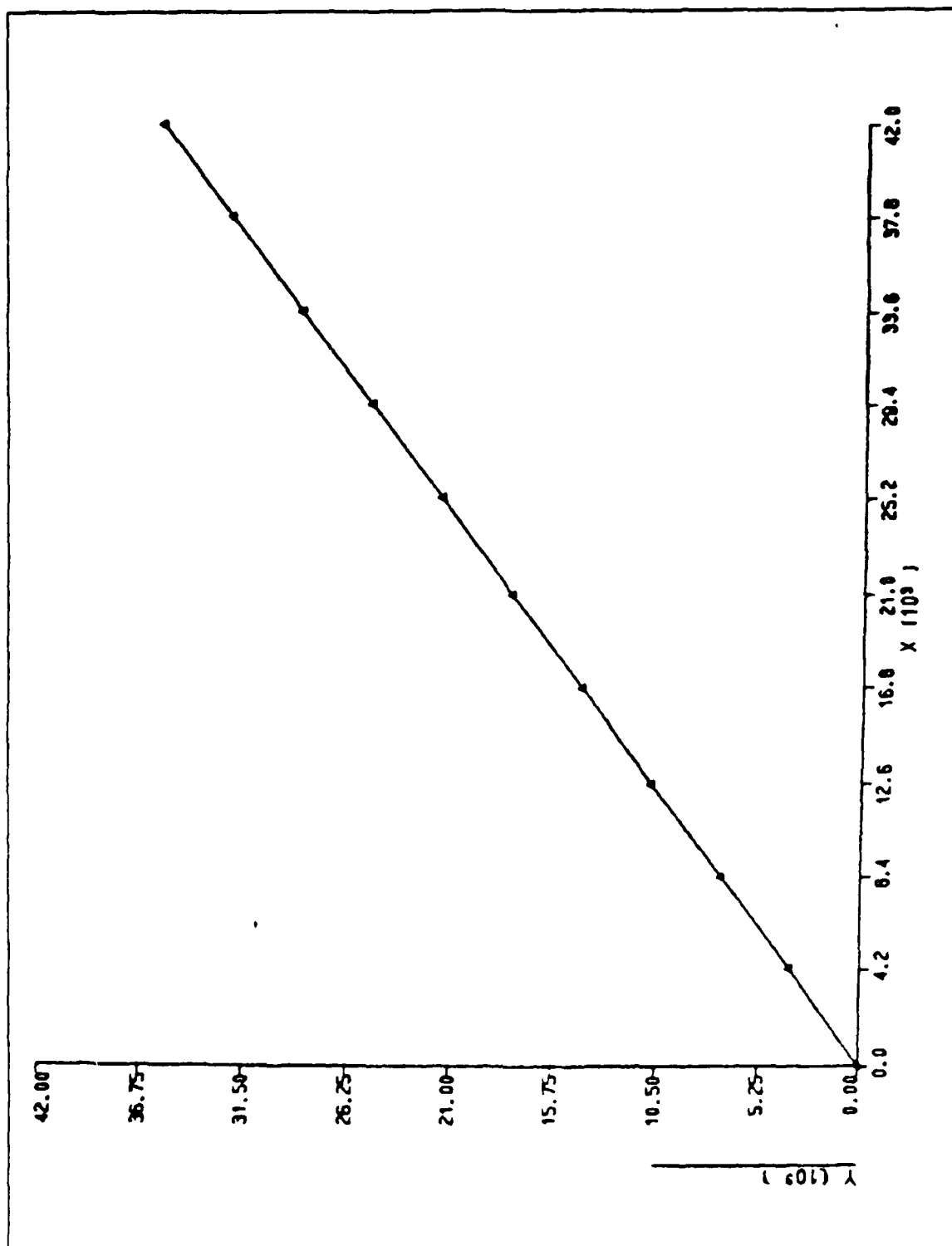


Figure 9. Intended and Actual Vehicle Tracks With No Added Bias

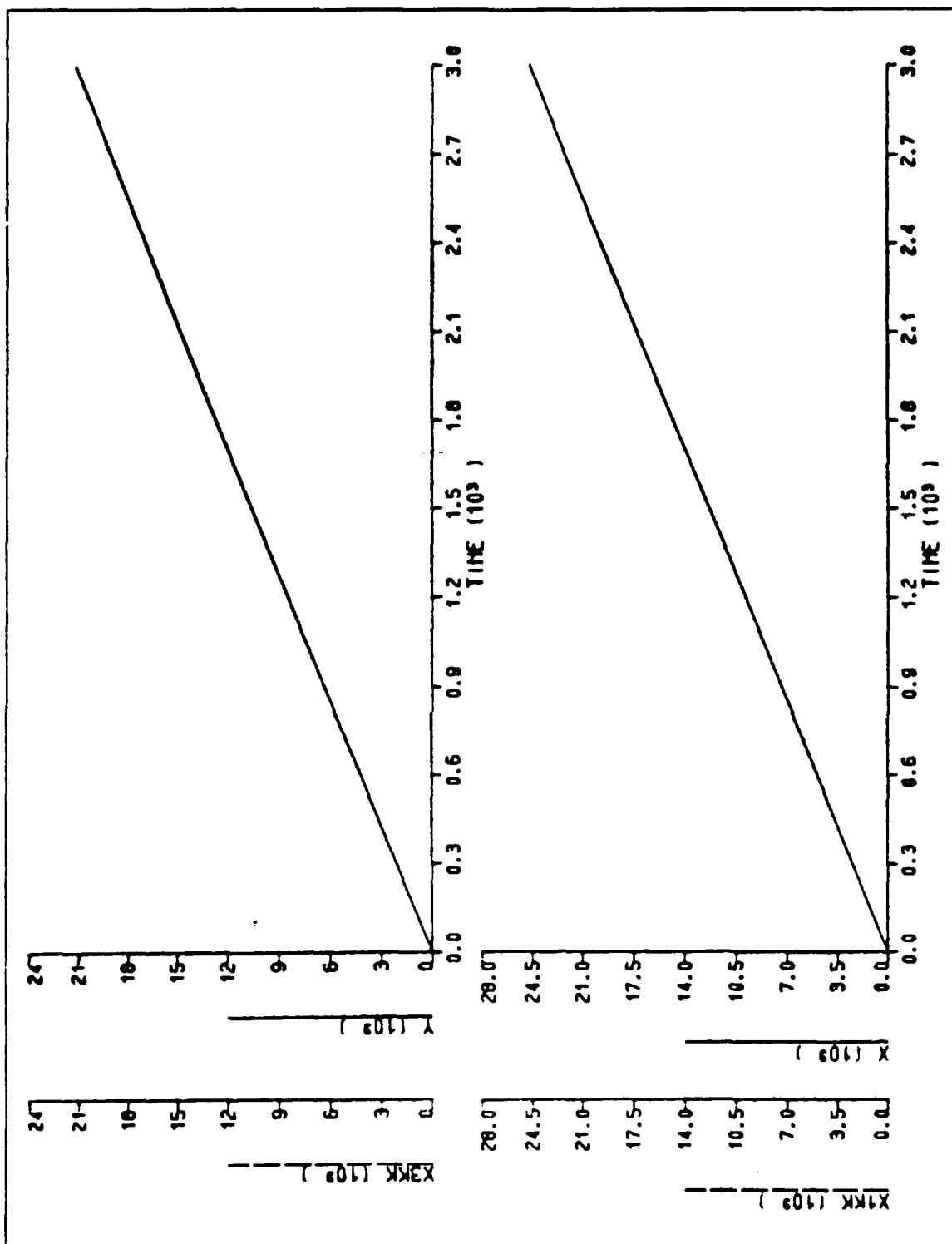


Figure 10. Position Coordinates With No Added Bias

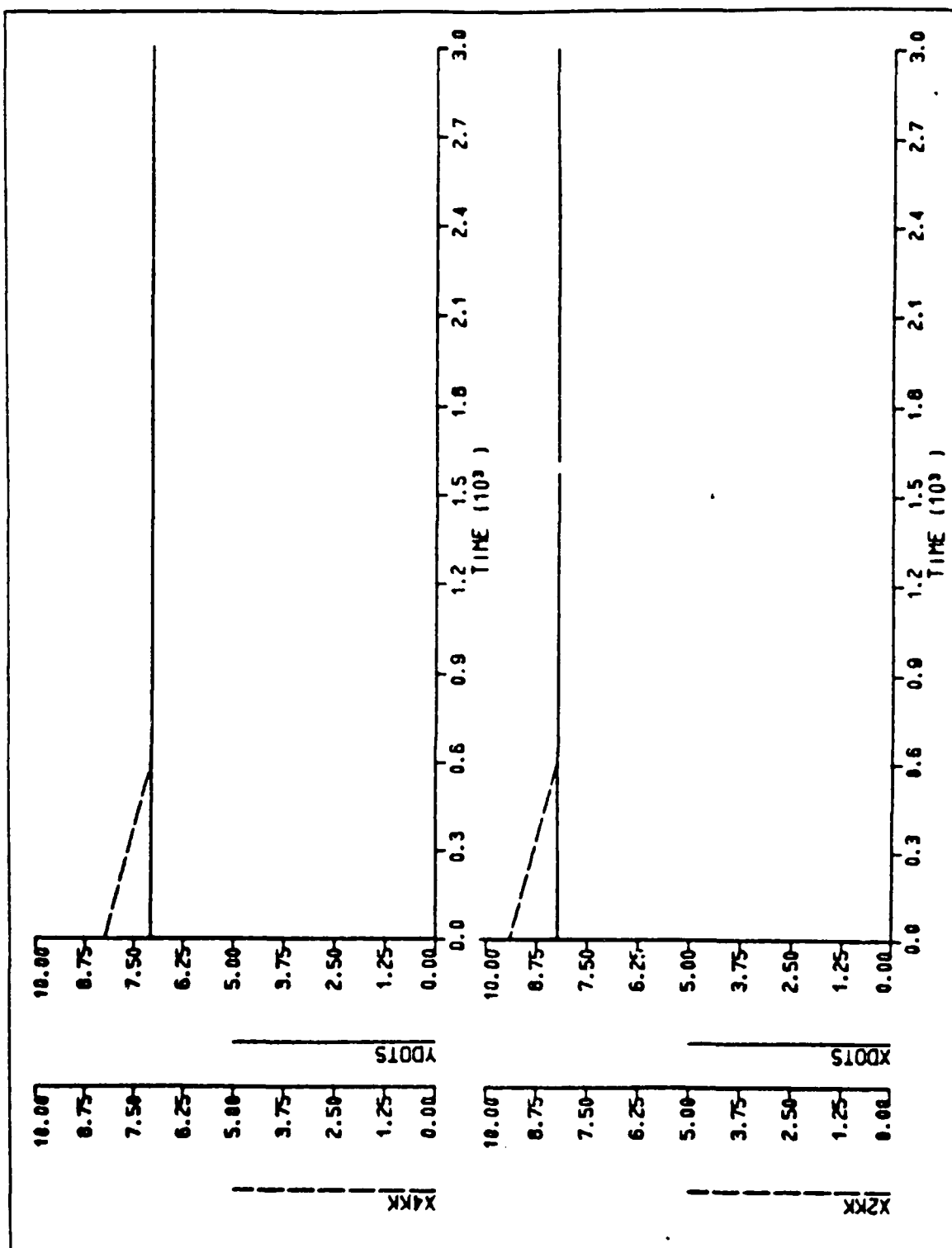


Figure 11. Velocity Components With No Added Bias

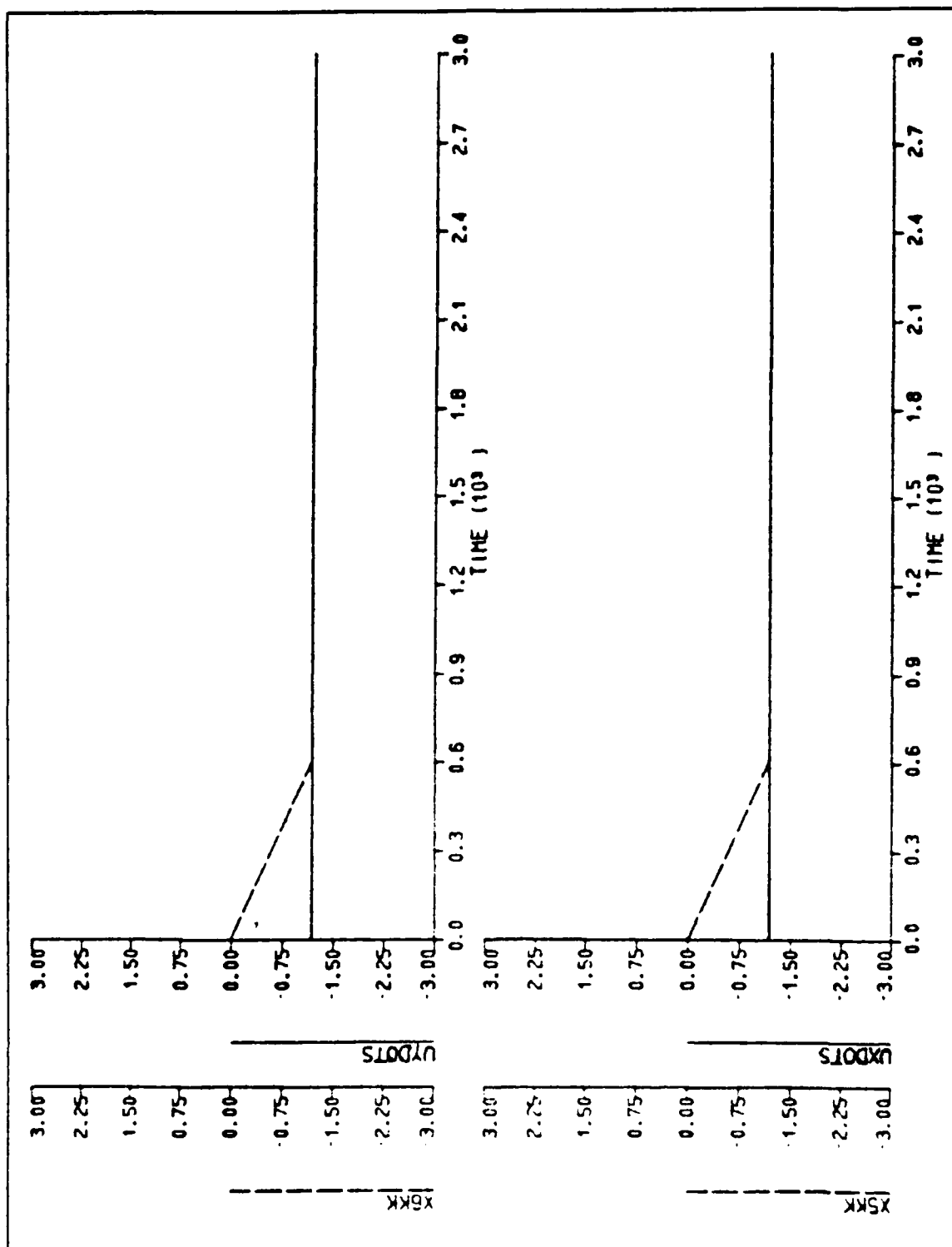


Figure 12. Set and Drift Components With No Added Bias

C. STATE ESTIMATION WITH BIAS AND COURSE CORRECTION

During this simulation, the filter's ability to maintain track during a course change was determined. A bias of .2 radians was added to the drift and .5 knots was added to the Set. After the filter determined an estimate for the Set and Drift components, the simulation program added them to a new desired course to arrive at the new heading command and updated $\hat{x}_{k-1|k}$. The new desired course was calculated by comparing the vehicle's current position to its destination and subtracting the respective components. As was previously done, the length of the run was 6000 seconds. The results are displayed in Figure 13 on page 29, Figure 14 on page 30, Figure 15 on page 31, and Figure 16 on page 32.

The figures demonstrate the program and filter's ability to correctly determine the overall Set and Drift and give a correct heading command to guide the vehicle to its destination. As can be seen, the filter successfully acquires u_x and u_y after only one sample and maintains a lock there after. This allows the vehicle to stay on course without having to constantly make corrections due to bad estimates from the filter. Next, the filter's ability to handle multiple Set and Drift changes will be tested.

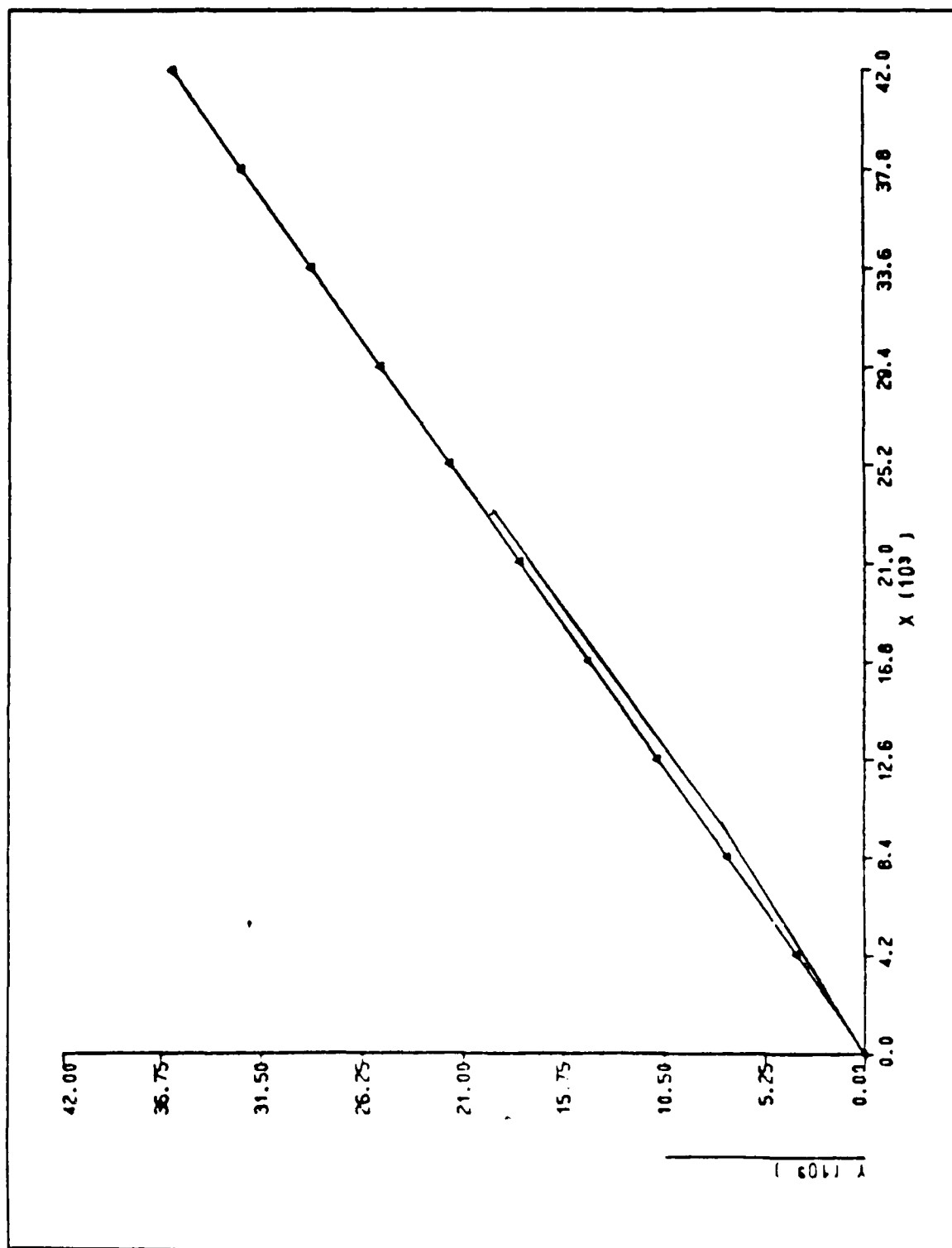


Figure 13. Intended and Actual Vehicle Tracks With Bias and Course Correction

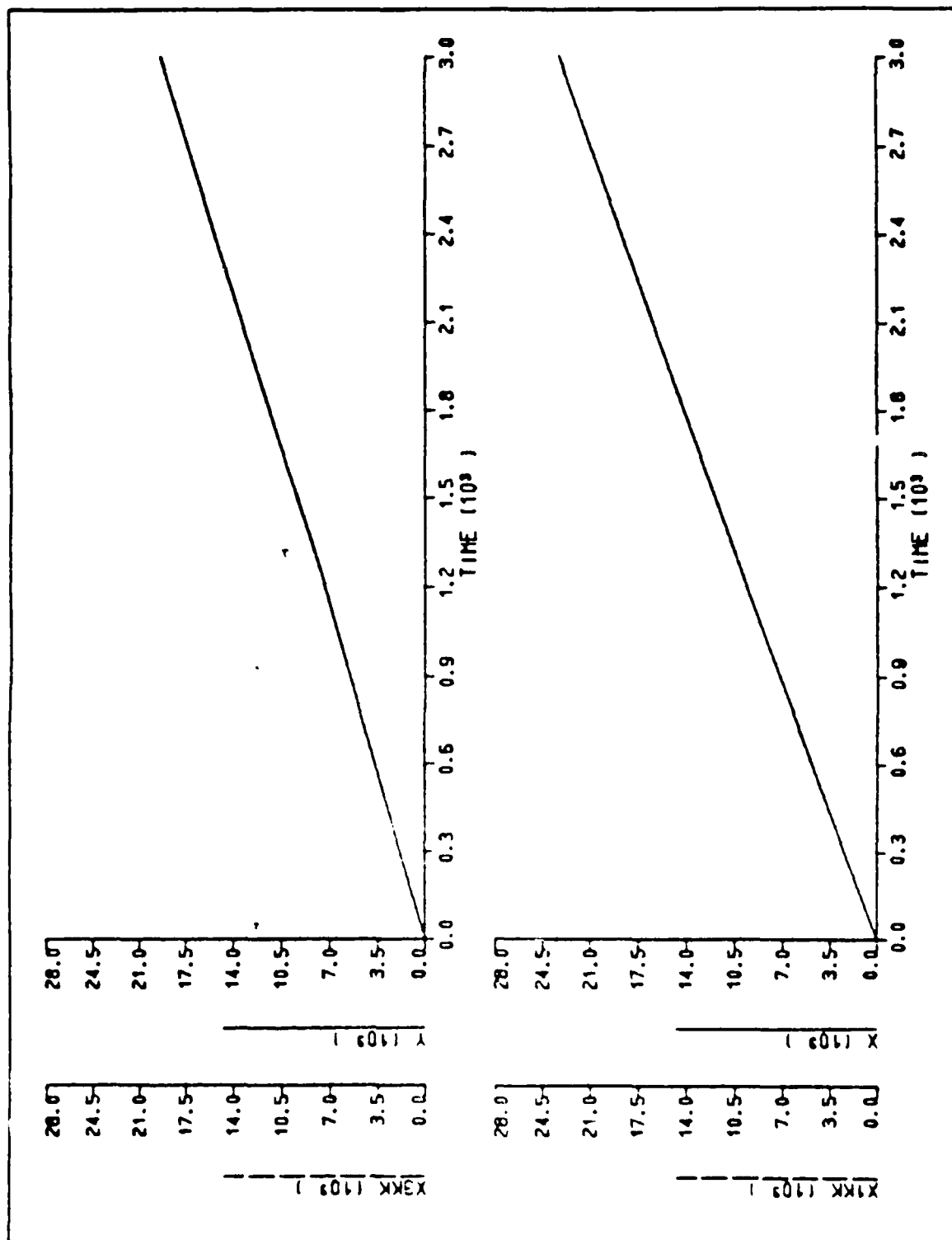


Figure 14. Position Coordinates With Bias and Course Correction

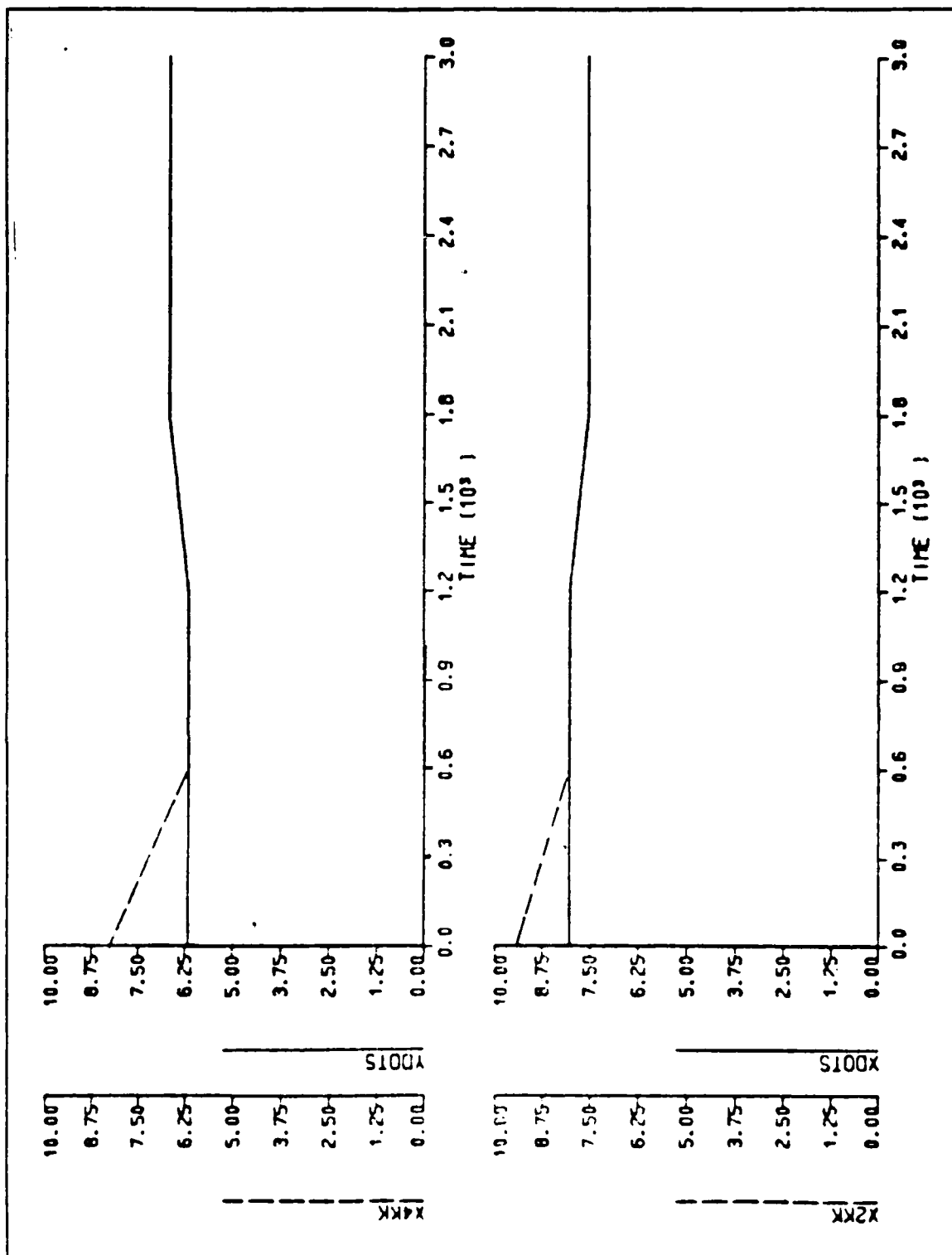


Figure 15. Velocity Components With Bias and Course Correction

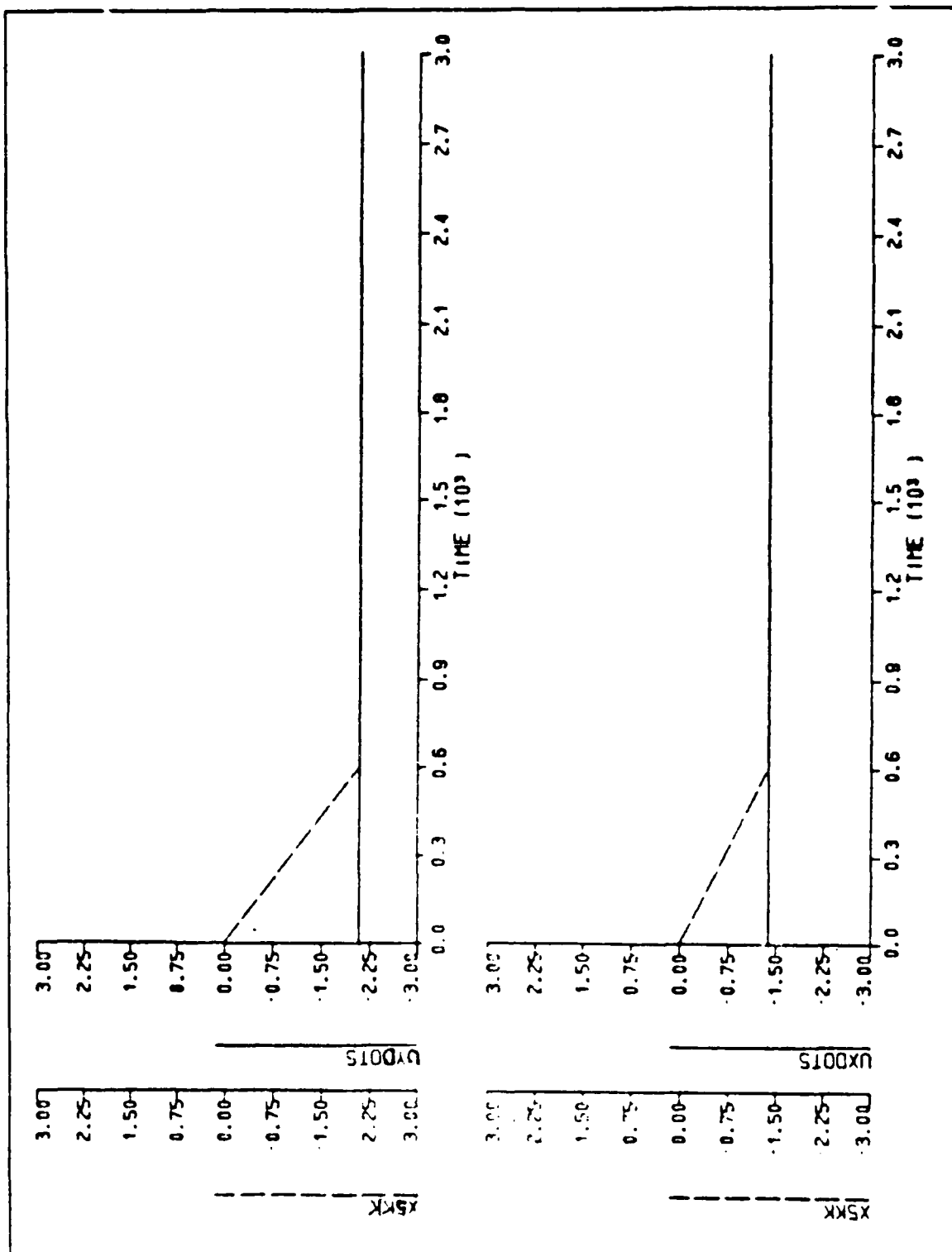


Figure 16. Set and Drift Components With Bias and Course Correction

D. STATE ESTIMATION WITH MULTIPLE SET AND DRIFT CHANGES

In this simulation, the Set and Drift was varied in a random manner at the beginning of every other sample. Values were created using the **Normal** function in DSL and added to the predicted Set and Drift. The **Normal** function randomly generates numbers that are distributed according to a Gaussian distribution function about a mean with a standard deviation. In this case the mean was zero and the standard deviation for the Drift was 7 degrees or .12 radians and .2 knots for the Set. The run was for 4200 seconds in order to obtain multiple variations at the sampling period of 600 seconds. The results were plotted and are displayed in the same manner as the previous two simulations.

In Figure 17 on page 34, the vehicle's actual track is off the intended track by as much as 500 feet which could be critical in any actual mission performed by a vehicle equipped with this system as its only navigation source. The rest of the graphs as shown in Figure 18 on page 35, Figure 19 on page 36, and Figure 20 on page 37 show the filter's ability to track any deviations in the Set and Drift experienced by the vehicle with only one sample after the change.

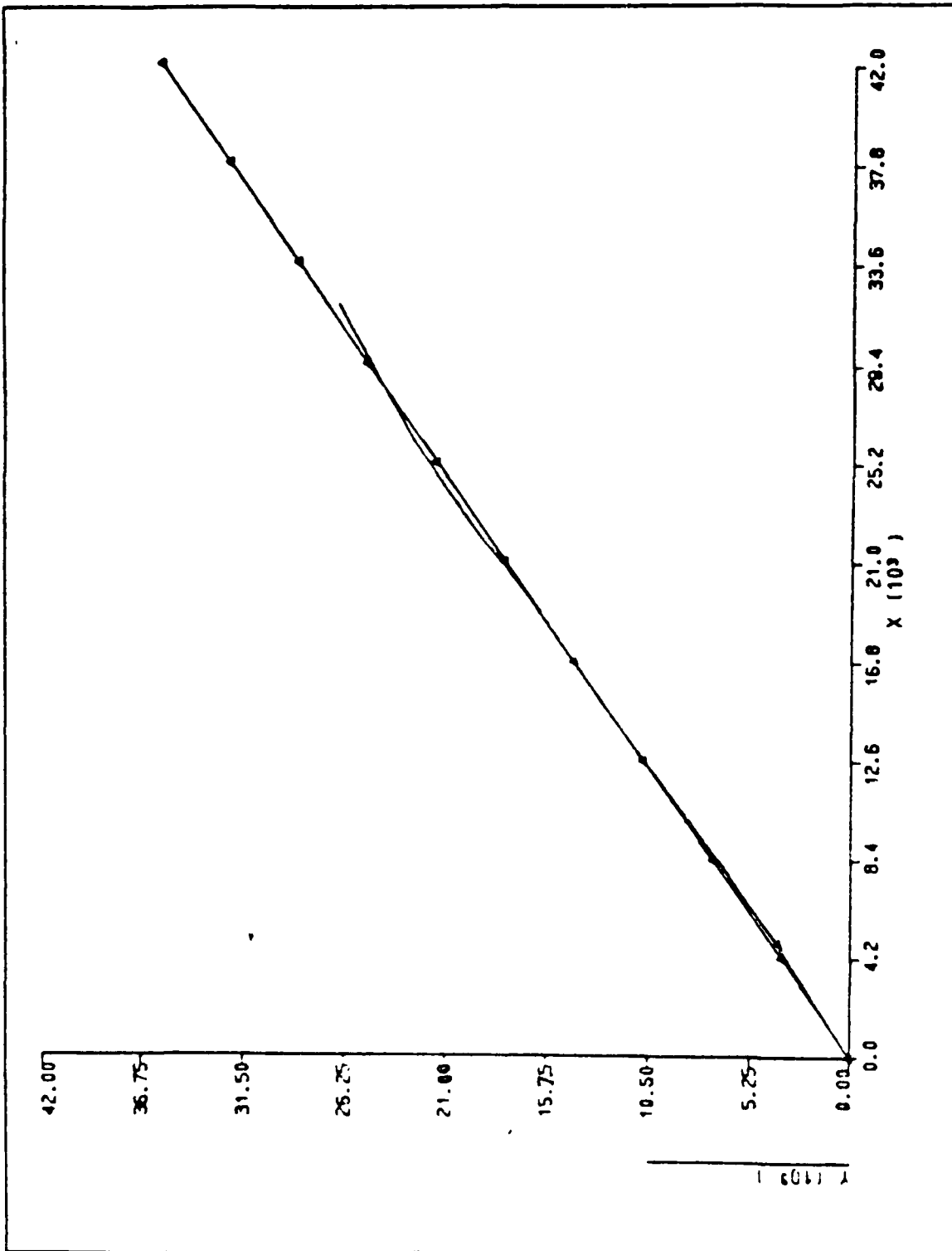


Figure 17. Intended and Actual Vehicle Tracks With Multiple Set and Drift Changes

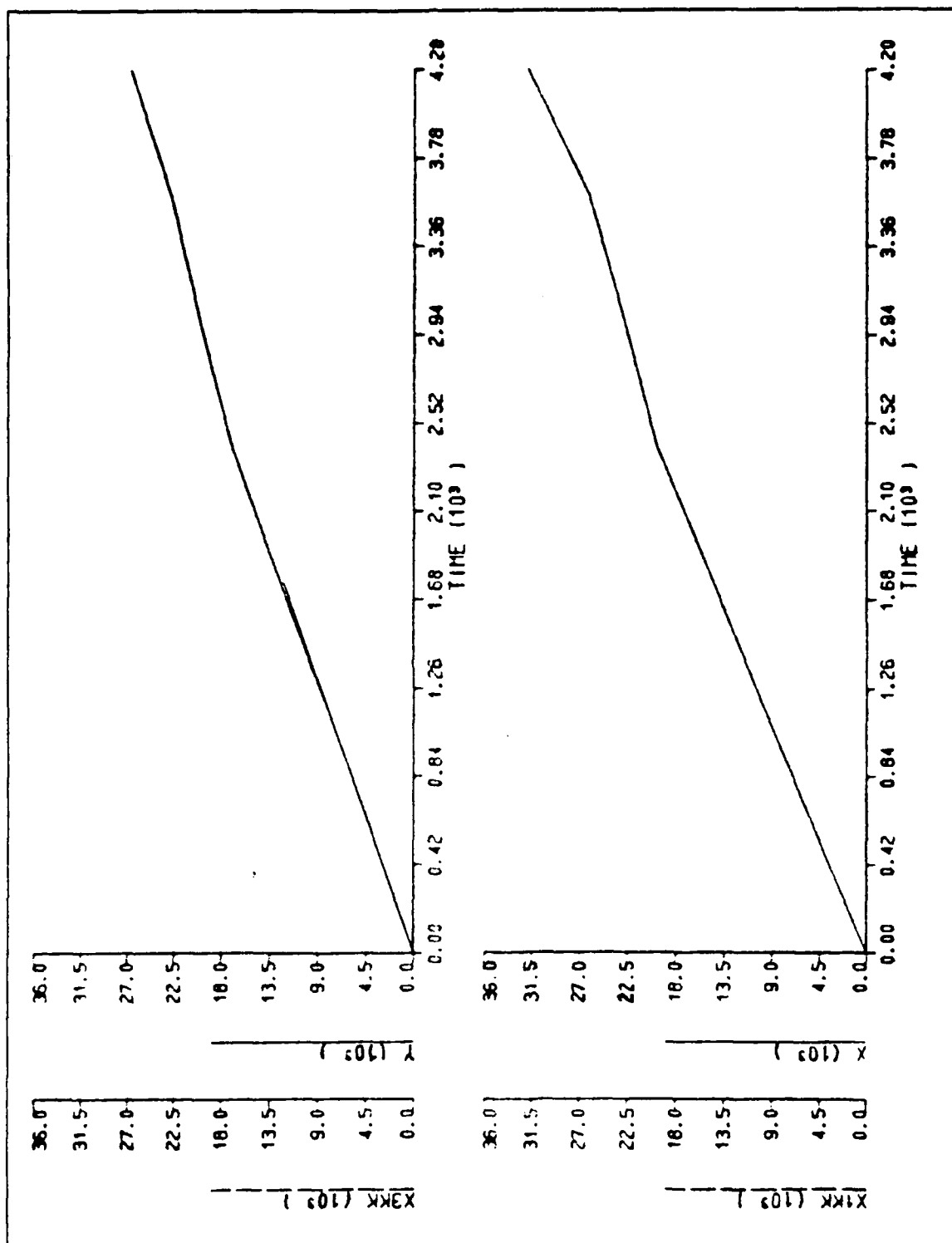


Figure 18. Position Coordinates With Multiple Set and Drift Changes

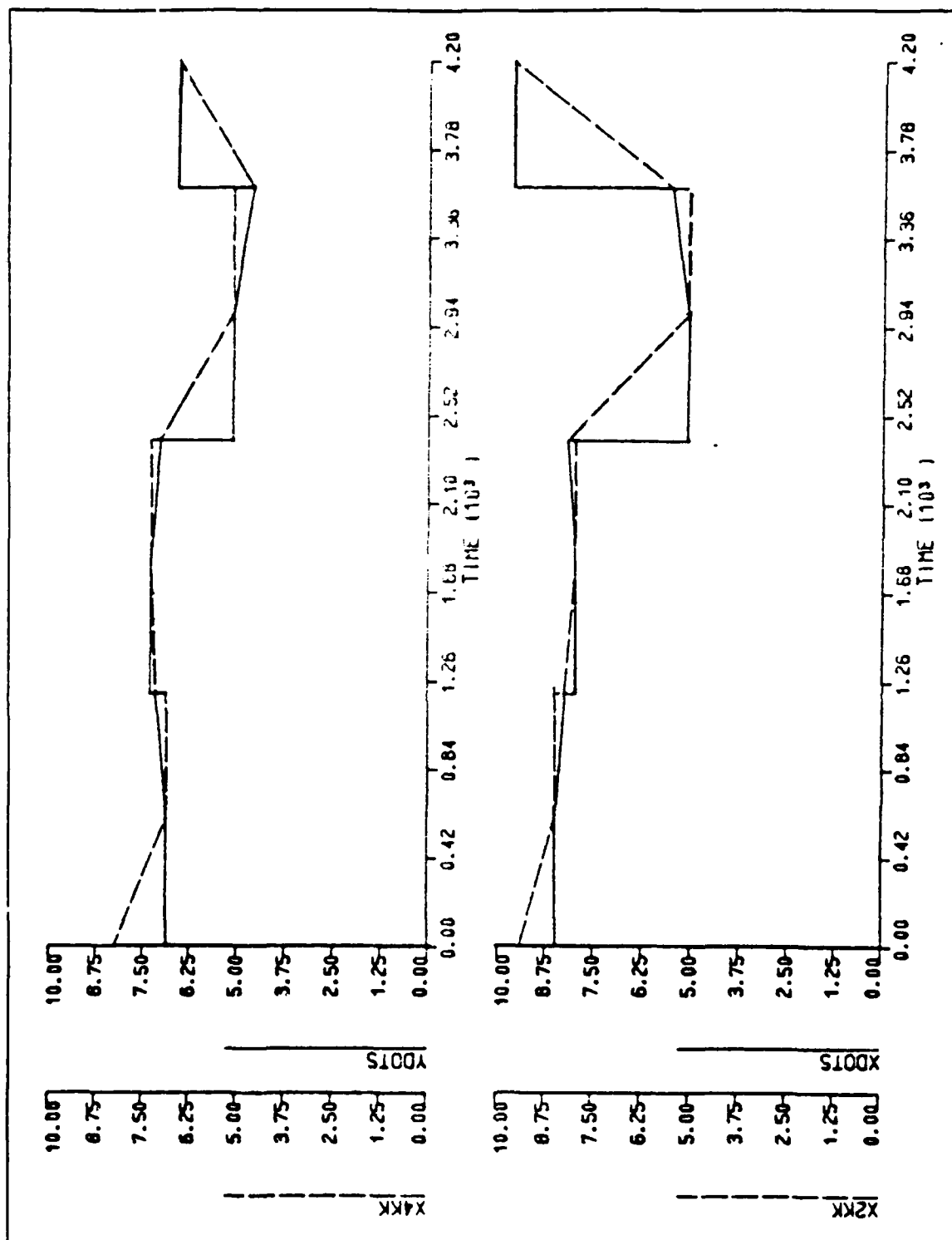


Figure 19. Components With Multiple Set and Drift Changes

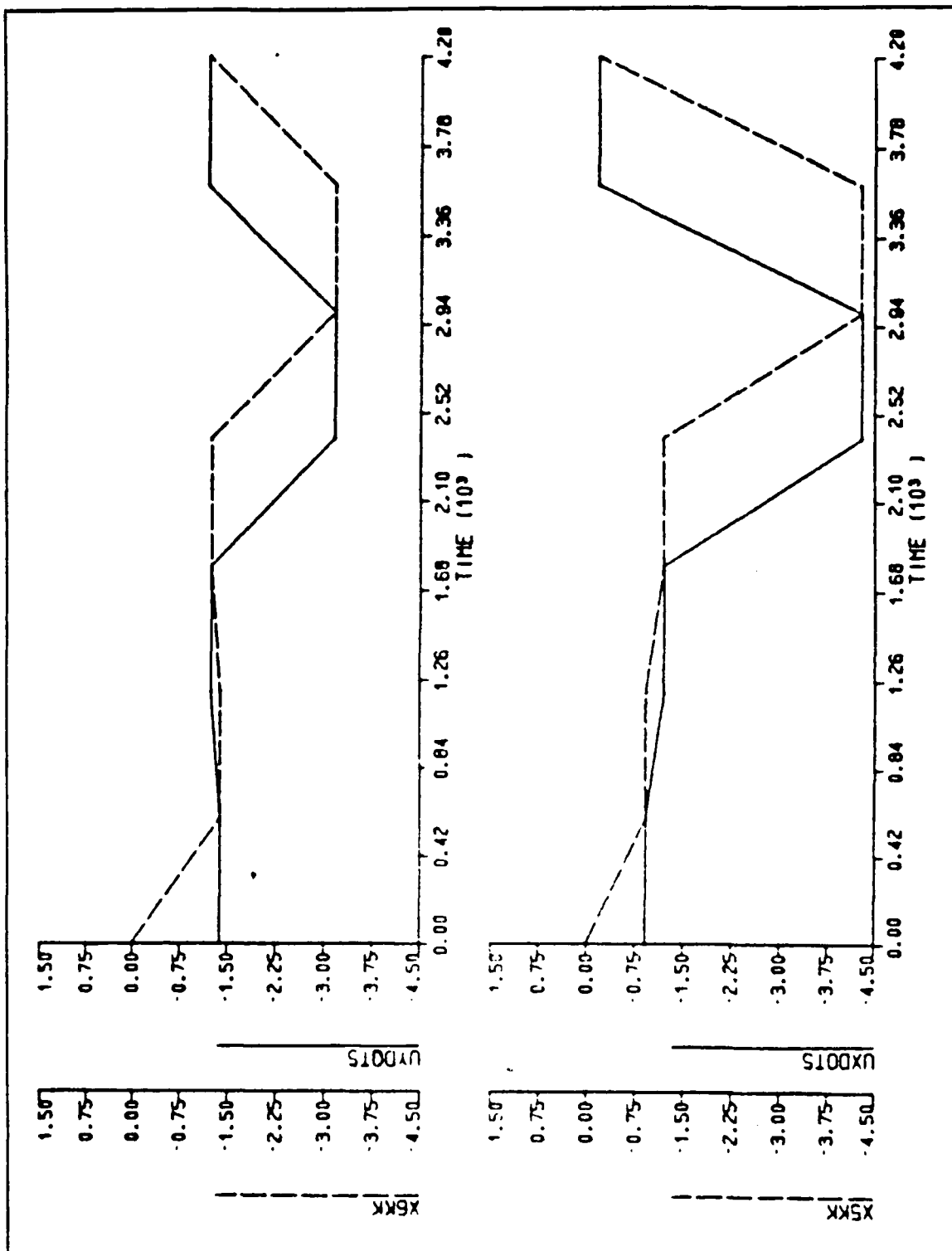


Figure 20. Set and Drift Components With Multiple Set and Drift Changes

E. STATE ESTIMATION WITH MEASUREMENT NOISE

In this simulation noise was added to the measurements of x , y , and ψ in order to determine how the Kalman Filter would respond. This noise was added using the Normal function described in the previous section. A different function was generated for each measurement with unique seed numbers, in order to preserve their independence, and zero mean about the actual value. The resulting graphs are displayed in Figure 21 on page 40, Figure 22 on page 41, Figure 23 on page 42, and Figure 24 on page 43. The run time was 4200 seconds and as in the previous simulation the Set and Drift were changed every other sample.

The graphs show the same results which were achieved in the third simulation with some small differences. The vehicle's intended and actual tracks are much closer than before which is due to a lesser amount of deviation in the Set and Drift variables. A close look at the plots of the state variables show some variation between their estimated and actual values which was not present before. These are the errors produced by the measurement noise. In order to see the actual differences, three tables were generated which show the actual and estimated values against time. The position coordinates, X and Y , are shown in Table 2 on page 39, \dot{x} and \dot{y} are in Table 3 on page 39, and \ddot{u}_x and \ddot{u}_y are in Table 4 on page 39.

From the tables, the errors range from 3 feet up to 21 feet for x and y , are less than .02 feet per second for \dot{x} and \dot{y} , and vary up to .18 feet per second for \ddot{u}_x and \ddot{u}_y . Note also that during the periods when there is no change in the Set and Drift, the estimates show a significant improvement. Errors such as these may be tolerable depending on the vehicle's assigned mission. The best way to minimize the error would be to take multiple GPS readings in order to obtain a better fix. But this would increase the amount of time that the vehicle would spend on the surface.

Table 2. DIFFERENCES IN POSITION COORDINATES

TIME	X		Y	
	ACTUAL	ESTIMATED	ACTUAL	ESTIMATED
600.	5092.4	5098.4	4103.5	4131.1
1200.	10056.	10060.	8351.7	8368.0
1800.	14880.	14890.	12712.	12708.
2400.	19803.	19801.	16962.	16971.
3000.	24869.	24863.	21097.	21103.
3600.	29822.	29831.	25359.	25367.
4200.	34518.	34525.	29864.	29870.

Table 3. DIFFERENCES IN VELOCITY COMPONENTS

TIME	\dot{X}		\dot{Y}	
	ACTUAL	ESTIMATED	ACTUAL	ESTIMATED
600.	8.4862	8.5048	6.8423	6.8961
1200.	8.0398	8.0520	7.2664	7.2318
1800.	8.0398	8.0520	7.2666	7.2318
2400.	8.4441	8.1857	6.8931	7.1053
3000.	8.4441	8.4338	6.8931	6.8878
3600.	7.8266	8.2801	7.5081	7.1072
4200.	7.8266	7.8265	7.5081	7.5015

Table 4. DIFFERENCES IN SET AND DRIFT COMPONENTS

TIME	\dot{U}_X		\dot{U}_Y	
	ACTUAL	ESTIMATED	ACTUAL	ESTIMATED
600.	-.92089	-.90220	-1.3892	-1.3354
1200.	-1.1535	-1.1414	-1.2030	-1.4073
1800.	-1.1535	-1.1414	-1.2030	-1.2378
2400.	-.91472	-1.1730	-1.3932	-1.1811
3000.	-.91472	-.92496	-1.3932	-1.3985
3600.	-1.3426	-.88910	-.98751	-1.3884
4200.	-1.3426	-1.3428	-.98751	-.99409

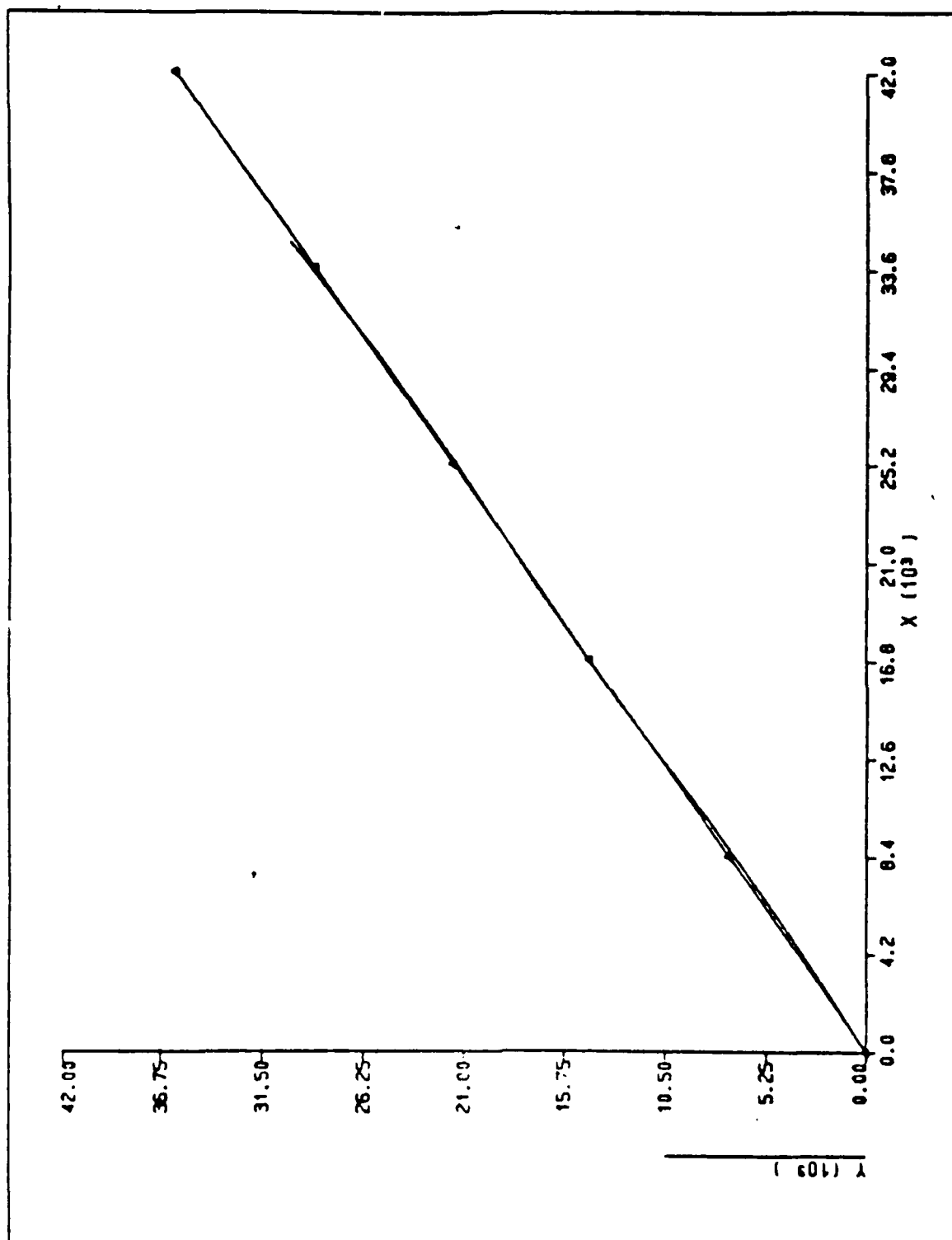


Figure 21. Intended and Actual Vehicle Tracks With Measurement Noise

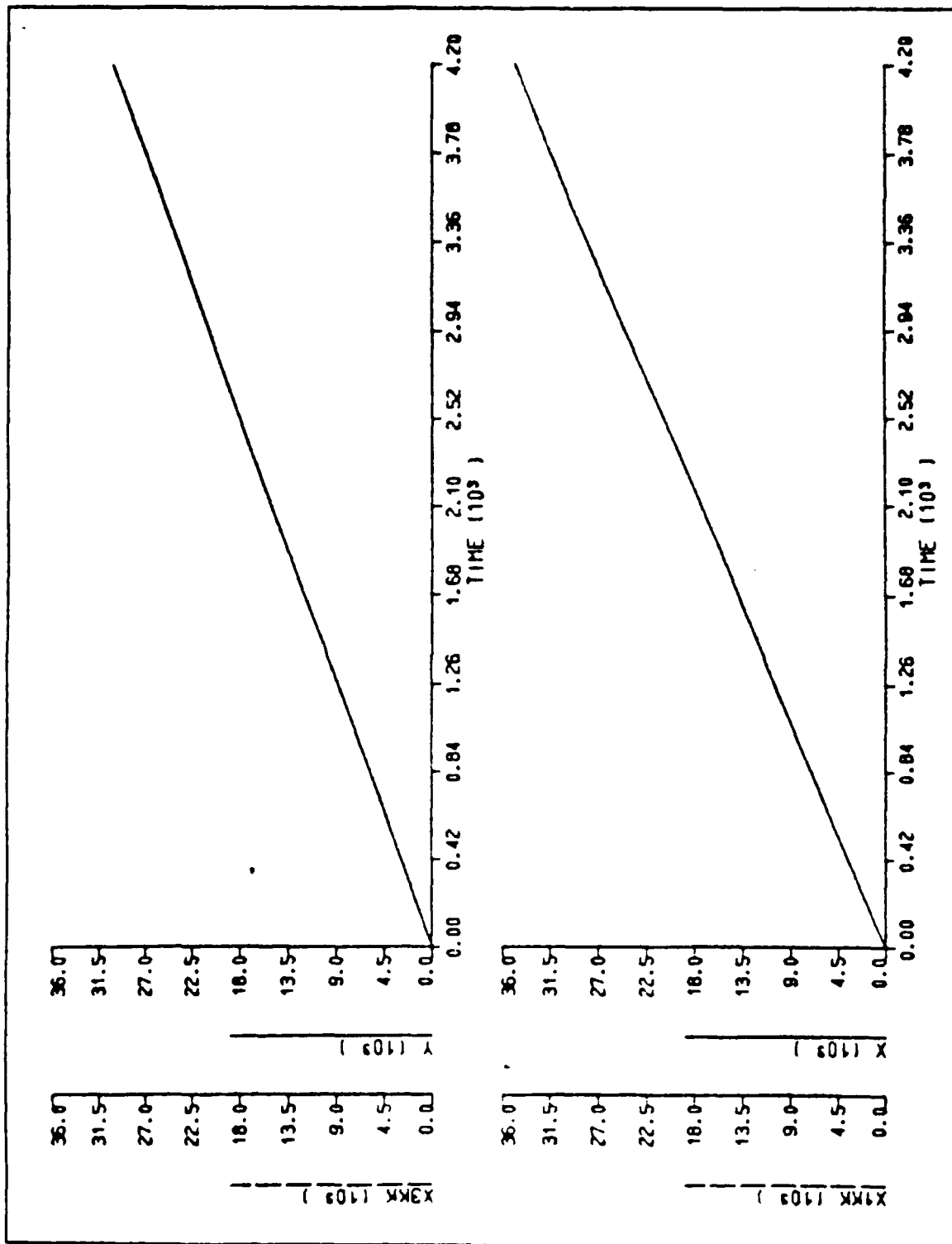


Figure 22. Position Coordinates With Measurement Noise

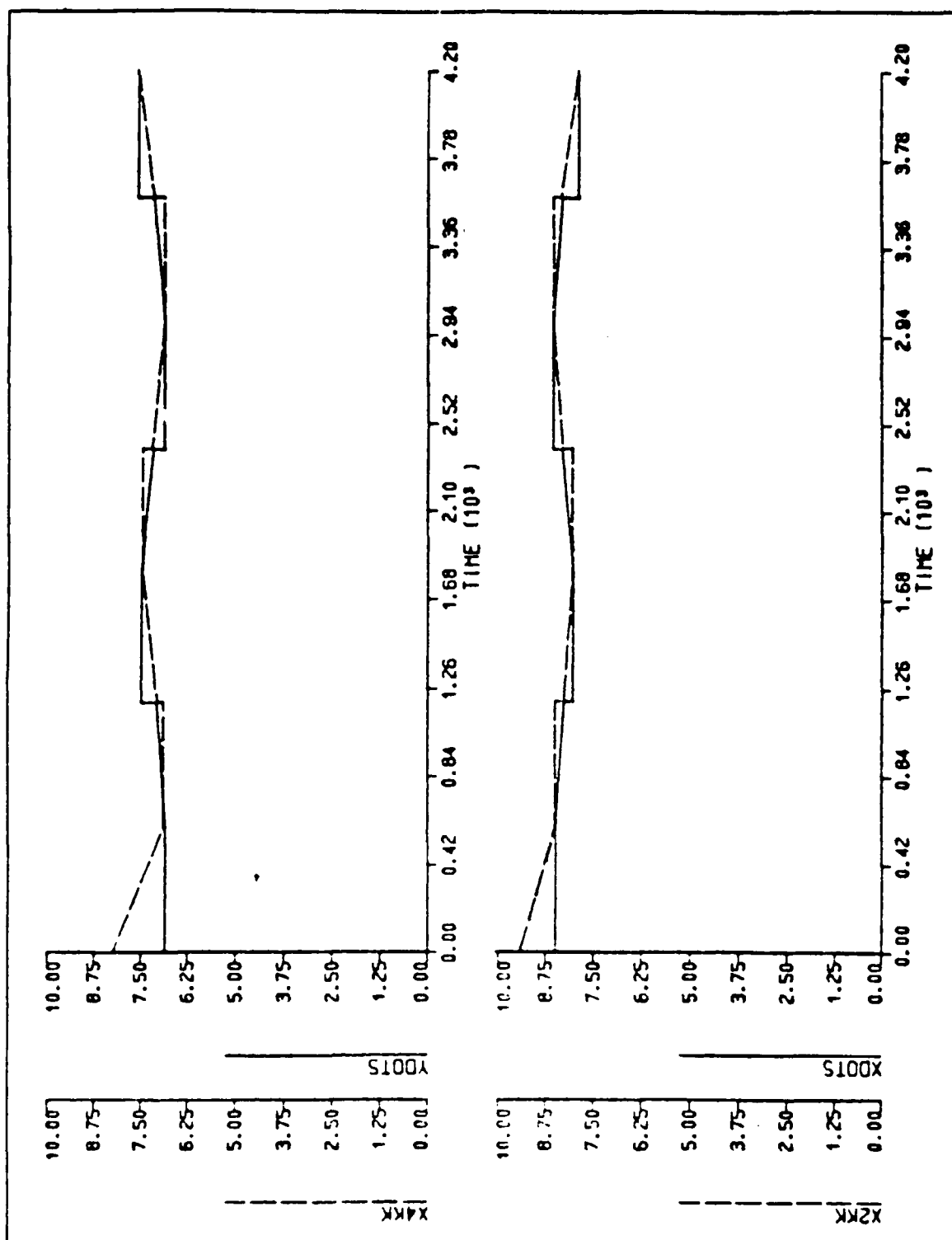


Figure 23. Velocity Components With Measurement Noise

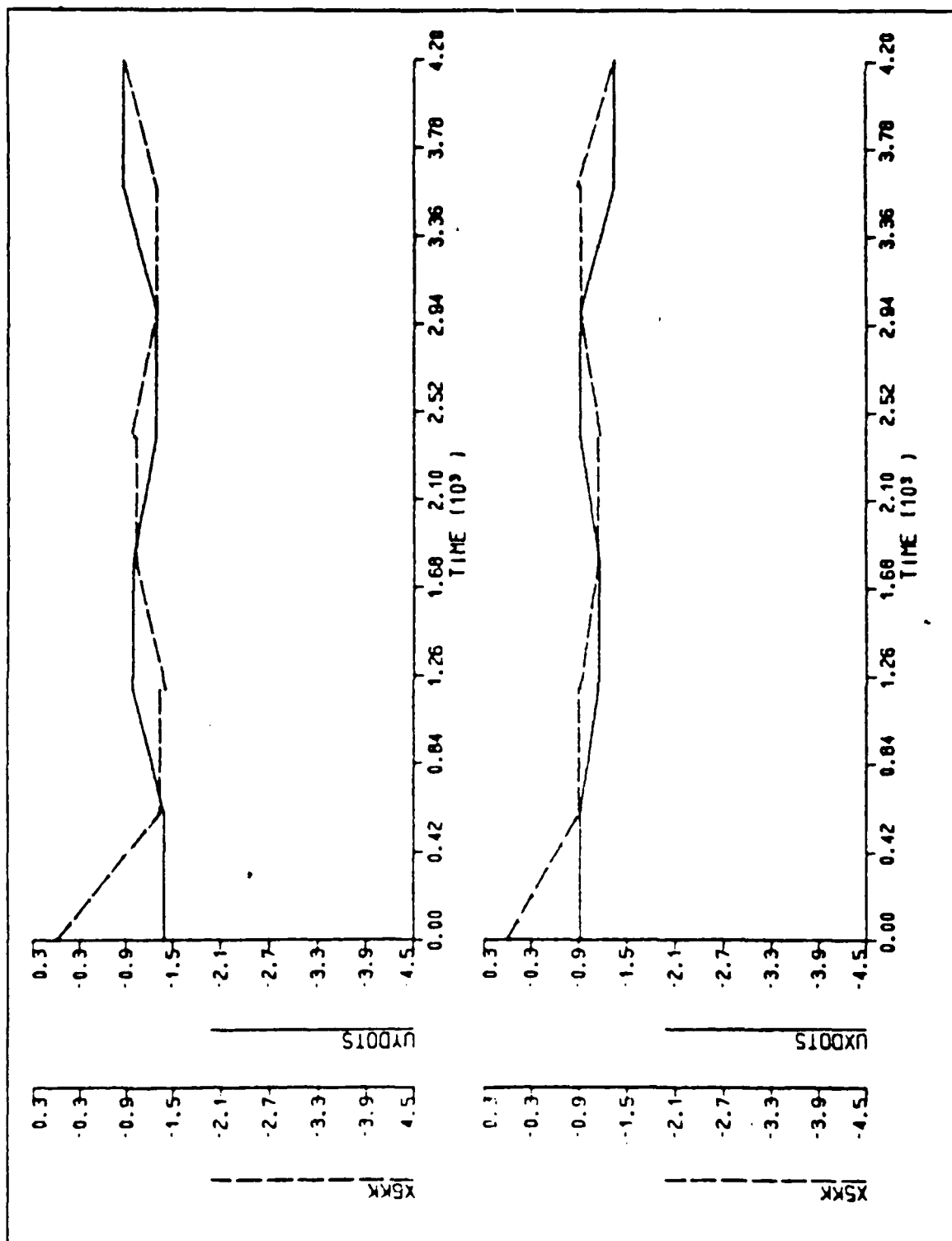


Figure 24. Set and Drift Components With Measurement Noise

V. CONCLUSION

A. ANALYSIS

The simulation results from Chapter 4 demonstrated the Extended Kalman Filter's ability to successfully identify the total Set and Drift that an underwater vehicle could experience on its journey. However, there are some limitations which have appeared. First, if the magnitude of the Set and Drift vary extensively between samples, then the vehicle's ability to navigate precise tracks is severely limited. There are three major factors which affect how far the vehicle will be driven off its intended course; the magnitude of the Set and Drift, the vehicle's ordered speed, and the time between samples. There is no control over the first factor but there is some latitude over which the operator can control. Increasing the speed and decreasing the sampling interval will allow less deviation from an intended track. However, both the resulting increase in power consumption and the increase in the probability of detection will have to be taken into account.

Another major limitation with using this system as a vehicle's sole navigation source is that the covert aspect of an underwater vehicle becomes lost when the GPS fixes are taken. Because of the frequencies at which GPS operates have very little ability to travel through water, the vehicle's antenna must break through the water's surface. To minimize the risk of detection, a system would have to be designed which would allow the antenna to be raised and lowered from a safe depth. An alternative would be to use the Omega Navigation System, whose frequencies can be reach down to about 40 feet, as a prime source and use GPS as an update at regular intervals or whenever the Omega data is suspect. This technique could allow the vehicle to refrain from surfacing as much and at a rate comparable to that required in updating an IMU system.

B. APPLICATIONS

1. Back up to an IMU

The most obvious place for this design is as a backup to an IMU system onboard an underwater vehicle which would take over in event of a failure. Current design philosophy seems to be centered on using conventional dead reckoning systems in which the angles and components are calculated using trigonometric functions. These type of complex calculations are difficult for a microprocessor to accomplish and machine roundoff errors can occur if the angles are close together. This Extended Kalman

Filter design requires only one trigonometric function [Eq. 3.15] and the matrix calculations are ideally suited for a microprocessor.

2. Mine Location

Because this system is lightweight, inexpensive, and has very little material which will disturb a magnetic field, a large fleet of plastic mine locating escorts could be developed. These escorts could lead ships through a mined area by locating suspected mine positions. As such, the vehicle would be composed of a body, a propulsion unit, a simple attitude control section, and a detector. This detector would be the most expensive component and it could take the form of a high resolution sonar or a magnetic anomaly detector. The shape of the vehicle could be designed that would reduce any pressure waves which would set off any pressure sensitive mines. The antenna system would not have to be as complicated as mentioned previously because the need to operate covertly would be erased. The communication link could be fiber optic and go to a mother ship or a group of mother ships working together.

3. Long Range Reconnaissance

A variant of the mine locating vehicle discussed above, would be as a long range reconnaissance vehicle with the mine detector replaced with some other sensor such as a communications receiver or an infrared detector. The missions would be limited to those which do not require precise navigation such as *coast or ship surveillance*. The advantages to using this system is that the power needed to run an IMU would be eliminated and the costs significantly reduced. Communication could be in the form of fiber optics for realtime information. Radio or data storage systems could be used for non-realtime. Delivery and retrieval could be accomplished through the use of a submarine, a ship, or small boats.

C. RECOMMENDATIONS FOR FUTURE RESEARCH

The following areas present valid opportunities for useful expansion of this work:

- Develop an antenna and support system, in order to receive the GPS signals in the presence of low to medium sea state and allow the vehicle to remain at depth.
- Build a prototype Extended Kalman Filter and test it as a back up on an existing autonomous underwater vehicle.

APPENDIX A. SIMULATION PROGRAM

This program simulates an underwater vehicle traveling through the water and is written in DSL. The Dynamic section mimics the vehicle's true dynamics, which were modeled in Chapter 2 as a second order system, and thereby calculates the vehicles true position and heading angle. The Sample section samples the system's position and heading angle every T seconds and these measurements are sent to the subroutine Kalman which estimates the states. After the estimates are returned, the program calculates a new heading command and updates the predicted state vector XP.

```

*
*
*
TITLE    VEHICLE SIM WITH KALMAN CORRECTION TO DEST / VAR F AND BIAS
PARAM    VEL=7.5,FVEL=1.,FANG=45., N=10,N1=7 ,N2=13 ,N3=3
PARAM    AX=0. ,AY=0. ,BX=14.0,BY=12.0
PARAM    SIGCA=0.3,SIGCV=0.3 ,SIGX2=100. ,SIGY2=100. ,SGYAW2=.1 ,...
          SIGCY=.1 ,SIGCX=.1,SIGX=10. ,SIGY=10. ,SIGYAW=.01
D        DIMENSION WAPNTX(0:10),WAPNTY(0:10)
D        DIMENSION F(6,6),XHAT(6),XP(6),Y(3),Q(2,2),R(3,3),P(6,6),H(3),
D      1  HP(3,6),G(6,2),XM(6)
EXCLUDE  IER,F,XHAT,XP,Y,Q,R,P,H,HP,G,XM
D        DATA F,XHAT,HP,Y,Q,R,P,H,XP,G / 133 *0. /
FIXED    I,J,N,N1,N2,N3
INITIAL
          A1=2.
          T1=5.
          A=0.
          TIMEW=0.
          T=DELS
*
*  ADDITION OF ERROR TO PREDICTED CURRENT VELOCITY AND ANGLE
          FANGS=(FANG+180.)*DEGRA + .2
          FVELS=FVEL
*
          FVELS=(FVELS*6000.)/3600.
          FVEL=(FVEL*6000.)/3600.
          VEL=(VEL*6000.)/3600.
          BX=BX*6000.
          BY=BY*6000
          UXDOTS=FVELS*COS(FANGS)
          UYDOTS=FVELS*SIN(FANGS)
          DC=ATAN2(BY-AY,BX-AX)
          DIST=SQRT((BX-AX)**2+(BY-AY)**2)
          INT=DIST/10.
          DO 10 I=0,10
            WAPNTX(I)=I * INT*COS(DC)
            WAPNTY(I)=I * INT*SIN(DC)
            WX=WAPNTX(I)
            WY=WAPNTY(I)

```



```

        CALL SAVE(S3)
10      CONTINUE
*      INITIALIZE  KALMAN MATRICIES
*
*      INITIALIZE  PHI
*
        DO 11 I=1,6
          F(I,I)=1.
11      CONTINUE
        F(1,2)=T
        F(3,4)=T
*
*      INITIALIZE Q
*
        Q(1,1)=SIGCX
        Q(2,2)=SIGCY
*
*      INITIALIZE GAMMA1
*
        G(1,1)=T
        G(2,1)=1.
        G(3,2)=T
        G(4,2)=1.
        G(5,1)=1.
        G(6,2)=1.
*
*      INITIALIZE R
*
        R(1,1)=SIGX2
        R(2,2)=SIGY2
        R(3,3)=SGYAW2
*
*      INITIALIZE XHAT WITH PREDICTED VALUES OF CURRENT COMPONENTS
*
        XHAT(5)=0.
        XHAT(6)=0.
*      XHAT(5)=FVEL * COS((FANG+180.)*DEGRA)
*      XHAT(6)=FVEL * SIN((FANG+180.)*DEGRA)
*
*      INITIALIZE HP
*
        HP(1,1)=1.
        HP(2,3)=1.
*
*
*      DETERMINATION OF ACTUAL X AND Y COORDINATES(SUNS INPUT)
*
DERIVATIVE
        HDGERR=YAW - HDG
        K=-4.41*HDGERR
        YAWDOT=REALPL(0.,.25,K)
        YAW=INTGRL(0.,YAWDOT)
        XDOT=VEL*COS(YAW)
        XDOTS=XDOT+UXDOTS
        YDOT=SIN(YAW)*VEL
        YDOTS=UYDOTS+YDOT

```

```

Y1=INTGRL(0.,YDOTS)
X=INTGRL(0.,XDOTS)
SAMPLE
IF(TIME.EQ.0.)THEN
  FANG=FANG * DEGRA
  IF(SIN(FANG)*SIN(DC).GT.0)THEN
    ALPHA=PI-ABS(FANG)+DC
    BETA=ASIN(FVEL*SIN(ALPHA)/VEL)
    HDG=DC + BETA
  ELSE
    ALPHA=PI-ABS(FANG)-DC
    BETA=ASIN(FVEL*SIN(ALPHA)/VEL)
    HDG=DC - BETA
  ENDIF
*   HDG=DC
  XHAT(2)=VEL*COS(HDG)
  XHAT(4)=VEL*SIN(HDG)
  XP(2)=VEL*COS(HDG)
  XP(4)=VEL*SIN(HDG)
  Y(3)=HDG
  DO 25 I=1,6
    XM(I)=XP(I)
25  CONTINUE
  CALL KALMAN(A,XHAT,F,XP,Y,Q,R,P,H,HP,G,IER)
  ELSE
    A=A + 1.
*   DETERMINATION OF REQUIRED COURSE TO DESTINATION
30  DWY=WAPNTY(10)- Y1
    DWX=WAPNTX(10)- X
    DC=ATAN2(DWY,DWX)
*   CALCULATE FORCE COMPONENTS,UANG AND UVEL
*
*   UPDATE H, Z, AND HP MATRICIES
*
  H(1)=XP(1)
  H(2)=XP(3)
  H(3)=ATAN2(XP(4)-XP(6),XP(2)-XP(5))
  DEN=(XP(2)-XP(5))**2 + (XP(4)-XP(6))**2
  HP(3,2)=(XP(6)-XP(4))/DEN
  HP(3,4)=(XP(2)-XP(5))/DEN
  HP(3,5)=-HP(3,2)
  HP(3,6)=-HP(3,4)
  Y(1)=X + NORMAL(N1,0,SIGX)
  Y(2)=Y1 + NORMAL(N2,0,SIGY)
  Y(3)=YAW+ NORMAL(N3,0,SIGYAW)
  DO 35 I=1,6
    XM(I)=XP(I)
35  CONTINUE
*
*   KALMAN FILTER UPDATE OF XHAT
*
  CALL KALMAN(A,XHAT,F,XP,Y,Q,R,P,H,HP,G,IER)
*
*   CALCULATION OF ESTIMATED CURRENT VELOCITY AND ANGLE FROM UPDATED XHAT
*
  UVEL=SQRT(XHAT(5)**2+XHAT(6)**2)

```

```

        UANG=ATAN(XHAT(6)/XHAT(5))
*
*   CALCULATION OF STEERED COURSES TO WAYPOINT AND DESTINATION *
*
        IF(A.EQ.A1-1.)THEN
        IF(SIN(UANG)*SIN(HDG).GT.0)THEN
            ALPHA=PI-ABS(UANG)+DC
            BETA=ASIN(UVEL*SIN(ALPHA)/VEL)
            HDGD=DC + BETA
        ELSE
            ALPHA=PI-ABS(UANG)-DC
            BETA=ASIN(UVEL*SIN(ALPHA)/VEL)
            HDGD=DC - BETA
        ENDIF
*
*   UPDATE X(K+1/K) FOR CRSE CHANGE
*
        XP(2)= VEL * COS(HDGD) + XHAT(5)
        XP(1)= T * XP(2) + XHAT(1)
        XP(4)= VEL * SIN(HDGD)+ XHAT(6)
        XP(3)= T * XP(4) + XHAT(3)
        HDG=HDGD
        ELSE
        HDG=HDG
        ENDIF
    ENDIF
DYNAMIC
    IF(A.EQ.A1)THEN
        BIAS=NORMAL(N,0.,.3)
        UXDOTS=FVELS*COS(FANGS+BIAS)
        UYDOTS=FVELS*SIN(FANGS+BIAS)
        A1=A1+2.
        T1=TIME + 2.
    ELSE
        UXDOTS=UXDOTS
        UYDOTS=UYDOTS
    ENDIF
    IF(TIME.LT.T1)CALL SAVE(S2)
    X1KK=XHAT(1)
    X2KK=XHAT(2)
    X3KK=XHAT(3)
    X4KK=XHAT(4)
    X5KK=XHAT(5)
    X6KK=XHAT(6)
    X1KKM1=XM(1)
    X2KKM1=XM(2)
    X3KKM1=XM(3)
    X4KKM1=XM(4)
    X5KKM1=XM(5)
    X6KKM1=XM(6)
40    IF(X.GE.BX) CALL ENDRUN
CONTROL FINTIM=9000.,DELS=600.
METHOD ADAMS
SAVE    (S1) 600.,X1KK,X2KK,X3KK,X4KK,X5KK,X6KK,...
        X1KKM1,X2KKM1,X3KKM1,X4KKM1,X5KKM1,X6KKM1
SAVE    (S2) 600.,X,Y1,UXDOTS,UYDOTS,XDOTS,YDOTS

```

```

SAVE      (S3)4000.,WX,WY
PRINT     600,HDG,X,Y1,UXDOTS,UYDOTS
GRAPH     (G1/S2,DE=TEK618)X(SC=8.4E3),Y1(SC=10.5E3,LO=0.)
GRAPH     (G2/S3,DE=TEK618,OV)WX(SC=8.4E3,AX=OMIT),...
          WY(MA=4,SC=10.5E3,AX=OMIT)
GRAPH     (G3/S2,DE=TEK618)TIME,X(LI=1,AX=OMIT,LO=0)
GRAPH     (G4/S1,DE=TEK618,OV)TIME,X1KK(LI=2,LO=0),X1KKM1(LI=3,LO=0)
GRAPH     (G5/S2,DE=TEK618)TIME,Y1(LI=1,AX=OMIT,LO=0)
GRAPH     (G6/S1,DE=TEK618,OV)TIME,X3KK(LI=2,LO=0),X3KKM1(LI=3,LO=0)
GRAPH     (G7/S2,DE=TEK618)TIME,XDOTS(AX=OMIT,LI=1,LO=0.,SC=1.8)
GRAPH     (G8/S1,DE=TEK618,OV)TIME,X2KK(LI=2,LO=0,SC=1.8),...
          X2KKM1(LI=3,LO=0.,SC=1.8)
GRAPH     (G9/S2,DE=TEK618)TIME,YDOTS(AX=OMIT,LI=1,LO=-1.5,SC=1.5)
GRAPH     (G10/S1,DE=TEK618,OV)TIME,X4KK(LI=2,LO=-1.5,SC=1.5),...
          X4KKM1(LI=3,LO=-1.5,SC=1.5)
GRAPH     (G11/S2,DE=TEK618)TIME,UXDOTS(AX=OMIT,LI=1,LO=-2.0,SC=.5)
GRAPH     (G12/S1,DE=TEK618,OV)TIME,X5KK(LI=2,LO=-2.0,SC=.5),...
          X5KKM1(LI=3,LO=-2.0,SC=.5)
GRAPH     (G13/S2,DE=TEK618)TIME,UYDOTS(AX=OMIT,LI=1,LO=-2.0,SC=.5)
GRAPH     (G14/S1,DE=TEK618,OV)TIME,X6KK(LI=2,LO=-2.0,SC=.5),...
          X6KKM1(LI=3,LO=-2.0,SC=.5)
LABEL (ALL) KALMAN CORRECTION TO DEST /BIAS= VAR/CUR CHG=2/MSMT ERROR/
END
STOP

```

APPENDIX B. KALMAN FILTER PROGRAM

This program is written in Fortran 77 and performs the necessary matrix calculations for the extended Kalman filter developed in Chapter 3. The routine calls three IMSLDP subroutines, LEQT1F, VMULFF, and VMULFP which accomplish the matrix multiplication operations. In order for Kalman to be compatible with DSL, it was written in double precision. After performing the calculations, this subroutine returns the predicted covariance matrix (P), the updated state estimate vector (XHAT), and the predicted state vector (XP) to the calling program.

```

C
      SUBROUTINE KALMAN (A,X,PHI,XKKM1,Z,Q,R,P,HK,HKP,GAMMA,IER)
C
      SPECIFICATIONS FOR ARGUMENTS
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER IER
      DIMENSION X(6),XKKM1(6),PHI(6,6),Z(3),Q(2,2),R(3,3),
1          HK(3),HKP(3,6),T1(6,6),T2(6,6),T3(6),P(6,6),
2          GAMMA(6,2),G(6,3)
      INTEGER I,J,N,NX,NY,L
      IER = 0
      N=6
      NY=3
      NX=1
      L=2
      K=A
      WRITE(6,2)((HKP(I,J),J=1,6),I=1,3)
      WRITE(6,5)(XKKM1(J), J=1,6)
      WRITE(6,6)(HK(J), J=1,3)
      WRITE(6,12)(Z(J), J=1,3)
C
      CALCULATE P IF K = ZERO
      IF (A .EQ. 0) THEN
      WRITE(6,3)((PHI(I,J),J=1,6),I=1,6)
      WRITE(6,7)((Q(I,J),J=1,2),I=1,2)
      WRITE(6,8)((GAMMA(I,J), J=1,2), I=1,6)
      CALL VMULFF ( GAMMA, Q, N, L, L,N,L,T2,N,IER)
      CALL VMULFP (T2, GAMMA, N, L, N,N,N, P,N,IER)
C
      DO 15 I=1,6
C
      P(I,I)=4000.
C 15 CONTINUE
      WRITE(6,13)((P(I,J),J=1,6),I=1,6)
      ELSE
      CONTINUE
      ENDIF
C
      CALCULATE GAIN MATRIX G(K)
      CALL VMULFP (HKP, P,NY, N, N,NY,N,T2,N,IER)
      CALL VMULFP (T2,HKP,NY, N,NY,N,NY,T1,N,IER)
      DO 35 I = 1,NY
      DO 30 J = 1,NY
      T1(I,J) = T1(I,J) + R(J,I)
30 CONTINUE

```

```

35 CONTINUE
   CALL LEQT1F(T1,N,NY,N,T2,0,T3,IER)
   IF (IER .EQ. 0) GO TO 40
   IER = 130
   GO TO 9000
40 DO 50 I = 1,NY
   DO 45 J = 1,N
      G(J,I) = T2(I,J)
45   CONTINUE
50 CONTINUE
   WRITE(6,9)((G(I,J),J=1,3),I=1,6)
C      CALCULATE P(K/K)
   CALL VMULFF (G,HKP, N,NY, N,N,NY,T2,N,IER)
   CALL VMULFF (T2, P, N, N, N,N,N,T1,N,IER)
   DO 75 I = 1,N
   DO 70 J = 1,N
      P(I,J) = P(I,J) - T1(I,J)
70   CONTINUE
75 CONTINUE
   WRITE(6,11)((P(I,J), J=1,6), I=1,6)
C      CALCULATE X(K/K)
   DO 60 I = 1,NY
      T3(I) = Z(I) - HK(I)
60 CONTINUE
   CALL VMULFF (G,T3, N,NY,NX,N,NY,T2,N,IER)
   DO 65 I = 1,N
      X(I) = XKKM1(I) + T2(I,1)
65 CONTINUE
   WRITE(6,4)(X(J), J=1,6)
C      CALCULATE P(K+1/K)
10 CALL VMULFF ( PHI, P, N, N, N,N,N,T1,N,IER)
   CALL VMULFF (T1, PHI, N, N, N,N,N, P,N,IER)
   CALL VMULFF ( GAMMA, Q, N, L, L,N,L,T2,N,IER)
   CALL VMULFF (T2, GAMMA, N, L, N,N,N,T1,N,IER)
   DO 25 I = 1,N
   DO 20 J = 1,N
      P(I,J) = P(I,J) + T1(I,J)
C      P(I,J) = P(I,J)
20   CONTINUE
25 CONTINUE
   WRITE(6,1)((P(I,J), J=1,6), I=1,6)
C      CALCULATE X(K+1/K)=XKKM1
   CALL VMULFF ( PHI, X, N, N,NX,N,N,T1,N,IER)
   DO 80 I = 1,N
      XKKM1(I) = T1(I,1)
80 CONTINUE
   GO TO 9005
9000 CONTINUE
   CALL UERTST (IER,6HFTKALM)
9005 RETURN
1  FORMAT(/,' PKP1K=' ,/,6(3X,F12.4,3X))
2  FORMAT(/,' ZP=' ,/,6(3X,F12.4,3X))
3  FORMAT(/,' PHI=' ,/,6(3X,F12.4,3X))
4  FORMAT(/,' XKK=' ,/,6(3X,F12.4,3X))
5  FORMAT(/,' XKKM1=' ,/,6(3X,F12.4,3X))
6  FORMAT(/,' HK=' ,/,3(3X,F12.4,3X))

```

```
7  FORMAT(/,' Q=' ,/,2(3X,F12.4,3X))
8  FORMAT(/,' GAMMA=' ,/,2(3X,F12.4,3X))
9  FORMAT(/,' KALMAN GAINS =' ,/,3(3X,F12.4,3X))
11 FORMAT(/,' PKK=' ,/,6(3X,F12.4,3X))
12 FORMAT(/,' Z=' ,/,3(3X,F12.4,3X))
13 FORMAT(/,' P/K=0 =' ,/,6(3X,F12.4,3X))
    END
```

LIST OF REFERENCES

1. Beaman, Ron, *Small Unit Navigation System Description and Deployment Concepts* Naval Ocean Systems Center, November 1983.
2. Walker Jr., Lindsey F., *Analysis and Simulation of an Autonomous Underwater Vehicle Control System*, Naval Postgraduate School, Masters Thesis, California. December 1983
3. Gelb, Arthur, *Applied Optimal Estimation*, The M.I.T. Press., 1986
4. Sorenson, H. W., *Parameter Estimation: Principles and Problems*, Marcel Dekker, Inc., 1980.
5. Kayton, Myron, *Avionic Navigation Systems*, John Wiley and Sons, New York, 1969.
6. Beyer, William H., *Standard Mathematical Tables 24th edition*, CRC Press, 1976.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Post Graduate School Monterey, CA 93943	1
4.	Professor Harold A. Titus, Code 62Ts Naval Post Graduate School Monterey, CA 93943	5
5.	Roberto Cristi, Code 62Cx Naval Post Graduate School Monterey, CA 93943	1
6.	Lt. James K. Easton Jr. 8445 Graves Ave. #24 Santee, CA 92071	3
6.	Mr. Ron Beaman Naval Ocean Systems Center Code 434 San Diego, CA 92152	1
7.	Mr. Richard Akita Naval Ocean Systems Center Code 434 San Diego, CA 92152	1
8.	Mr. James K. Easton Bldg. R-1, M/S C311 Hughes Aircraft Co. P.O. Box 92426 Los Angeles, CA 90009	1

END

DATE

FILMED

8-88

DTIC